Final Report

Title: Towards Next Generation WWW: Push, Reuse and Classification

Contract Number: FA5209-05-P-0253

AFOSR/AOARD Reference Number: AOARD-054033

AFOSR/AOARD Program Manager: Tae-Woo Park, Ph.D.

Period of Performance: 01 01 2005 – 30 06 2006

Submission Date: 04 08 2006

PI: Dr. Byeong Ho Kang/University of Tasmania
CoPI: Professor Paul Compton/University of New South Wales
Professor Hiroshi Motota/Osaka University

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **05 AUG 2006**	2. REPORT TYPE **Final Report (Technical)**	3. DATES COVERED **23-02-2005 to 30-06-2006**

4. TITLE AND SUBTITLE **Towards Next Generation WWW: Push, Reuse and Classification**	5a. CONTRACT NUMBER
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S) **Byeong Ho Kang**	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **University of Tasmania,GPO Box 252-100,Hobart TAS 7001,Australia,AU,7001**	8. PERFORMING ORGANIZATION REPORT NUMBER **AOARD-054033**
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) **The US Resarch Labolatory, AOARD/AFOSR, Unit 45002, APO, AP, 96337-5002**	10. SPONSOR/MONITOR'S ACRONYM(S) **AOARD/AFOSR**
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT
The major problem of the current WWW technology is that it is based on the ?pull? style of information delivery, where the uploaded information waits for visitors, and not the ?push? style (Franklin and Zdonik 1998), where the new information is delivered to the users when it becomes available. Although there are several research studies focusing on the development of ?push? based information delivery, these studies overlook following two important functions: Firstly, many of the new studies are not concerned with existing HTML documents. It is not wise to expect that all people will follow the new suggested representation like XML or RSS and will convert their existing information to the new format. Secondly, the information classification system is the other issue. Without using the appropriate classification system, people find that delivered information is often redundant. Therefore, an automated classification system that selects only the relevant information for each user is required. The main research task for this system is how to implement the incremental knowledge acquisition process for the classification knowledge because human classification knowledge is always heuristic and changes rapidly and, therefore, it is necessary to maintain the knowledge base incrementally.

15. SUBJECT TERMS
Data Mining, Knowledge Acquisition, Web Services

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**		**56**	

Table of Contents

1 Executive Summary

The main purpose of this project is the use of the Ripple Down Rules and advanced Web technologies to provide the push based Web information services. Although the Web technology has been favoured by many users and it is one of the largest Internet services, the limitations of the base technology hinder the developer from developing further services. The main problem of Web technology is that information providers can not push the information to clients but has to wait for the requests. This means that the clients should know where the information is and when the information is available. One main suggestion to overcome this problem is the introduction of 'Push" based information services such as RSS. However, there are two main problems in this approach. One is that it only handles XML documents and the other is the classic information overflow caused by providers.

This project proposes the integration of the flexible HTML processing technology and information classification technology by Ripple Down Rule method, the incremental knowledge acquisition method grown from the expert system area. In addition to the monitoring and classification technology, this project includes the flexible delivery system for collected information. The project has been motivated by the previous prototype system developed by MCRDR research group in University of Tasmania initially and two partners, Prof. Paul Compton (University of New South Wales) and Prof. Motodoa (Osaka University) joined this project. The current project is funded by AOARD from early 2005 and AOARD has organized the collaboration research with AFRL in ROME.

Three main PIs visited Air Force Research Lab, ROME in February, 2005 and introduced this project to the information research group in ARL leaded by Dr. John Salerno. Dr Kang (UTAS) and Dr. Park (AOARD) visited ARL in October 2005 and further discussion was held to set up a new project based on the current project. The new project "Personal Assistant for Web Searches: Multi-Tasking and Monitoring" has been approved by AOARD in middle of 2006 and we plan to continue the collaboration with ARL through the new project.

In the middle of 2005, the prototype system was introduced to the team who were maintaining the Tasmanian State Government Web Page http://www.service.tas.gov.au/. The project team set up the information tracking service by using one of the suggested modules in this project. This tracking service can monitor about 270 Australian Government Web pages every one or two hours and can collect new information to the local server. The service has been used for about one year and the system is still in routine service.

The project team also has tested more than 5 different general information domains for this system, health, IT, security, conference information and so on. A few papers has beenpublished and more papers are under progress as an outcome of this project.

The project team are very glad to complete this project with successful outcomes and continue the further project with AOARD. We greatly appreciate the program manager Dr. Tae Woo Park and staffs in AOARD for their support and help through out this project. Also, we also appreciate the warm hospitality of ARL while we were visiting ARL for the seminar and their support. We are very impressed with the current personal meta-search engine project in ARL and we think that there is good potential between the two teams.

2. Objectives

The aim of this project is to investigate new approaches for the development of 'push' based WWW information service systems in a specialized domain. We have been developed an intelligent Web information management system to test an idea of new approaches. The system consists of following three components:

- Web information monitoring (Monitoring Server): Monitors target web sites regularly.
- Information classification (MCRDR Classifier): Classifies collected information
- Information delivery (Information Delivery Server: IDS): Delivers information to users

The Monitoring Server revisits target Web sites periodically and automatically reports new information of those Web sites. This component supports automated and timely information gathering. MCRDR classifier supports automated document classification using Multiple Classification Ripple Down Rules (MCRDR), an incremental knowledge acquisition method. This approach is beneficial because even naïve user can incrementally build classification rule base without help from knowledge engineer. Lastly, the IDS can support information sharing by providing information via various methods. In this research we focus on the fully automated Web portal publication system by using the Monitoring Server and MCRDR Classifier. The main aim of this project is how the developed push technology can be used in a specialized domain and how to deliver the information effectively to different users.

3. System Report

1) System analysis and design

We divided the whole system into several sub-systems because it is easy to solve the problem in a small size and helps to check progress easily. Table 1 summarizes the requirements of the sub-systems.

Table 1. Sub-systems lists of the target system

Components	Sub-systems
Monitoring Server/ *iWebServer in User Manual*	• Web site registration sub-system: users can register Web sites to be monitored. • HTTP request generator sub-system: this module sends HTTP request to the target Web servers. • Monitoring Server scheduler sub-system: users can set up monitoring schedule
MCRDR Classifier/ *iWebClient in UserManaul*	• HTTP message analysing sub-system: this module parses HTTP response messages • Feature extracting sub-system: this module eliminates noise data in the HTML sources and represents documents features • Knowledge acquisition sub-system: this module helps to create rule base and maintain it without help from knowledge engineer, which is implemented with MCRDR knowledge acquisition algorithm.
IDS/*iWebPortal*	• Web portal generating sub-system: this module automatically generates Web portal based on the Monitoring Server or the MCRDR Classifier.

2) Implementations

- The Monitoring Server and the MCRDR Classifier are implemented with C++ programming language and MySQL database system and run on the Windows operating system. We separate Monitoring Server from MCRDR Classifier to handle multiple domains effectively.
- The IDS is implemented with PHP programming language and MySQL database system and run on the Window or Linux system with Apache Web server. The current design focuses on easy maintenance for Web publication.

3) Systems in routine use

- Systems are completed and demonstration sits is now under operation: http://www.comp.utas.edu.au/iWeb

- Systems are tracking Australian Government Web Sites Information (270 Web pages) for Tasmanian State Government Web Page (http://www.service.tas.gov.au/) http://www.comp.utas.edu.au/iweb/iPrj/librarynew/

- Systems are tracking IT news information from world wide newspapers for "The 19th Australian Joint Conference on AI 2006: http://www.comp.utas.edu.au/ai06" and "Pacific Knowledge Acquisition Workshop: http://www.comp.utas.edu.au/ai06.

4) Demonstration site is available from: http://www.comp.utas.edu.au/iWeb

5) System Setup Manual is included in Appendix and software systems are included in this report. Please note that the user requires to install Apache Web Server includingand MySQL (Ver 4.xxxx). The configuration of these severs can be different in various systems. If you have any problem to install included software, you can contact:

Dr. Tae Woo Park (AOARD)
Email: tae-woo.park@aoard.af.mil
Phone: +81 3 5410-4409

Dr. Byeong Ho Kang, University of Tasmania
Email: bhkang@utas.edu.au
Phone: +61 3 6226-2919

Please note that the names of software is used in the Manual is different from this document. iWebServer, iWebClient and iWebPortal is used in the user manual respectively "Mon Server", "MCRDR Classifier" and IDS.

4. Abstract

The major problem of the current WWW technology is that it is based on the 'pull' style of information delivery, where the uploaded information waits for visitors, and not the 'push' style (Franklin and Zdonik 1998), where the new information is delivered to the users when it becomes available. Although there are several research studies focusing on the development of 'push' based information delivery, these studies overlook following two important functions: Firstly, many of the new studies are not concerned with existing HTML documents. It is not wise to expect that all people will follow the new suggested representation like XML or RSS and will convert their existing information to the new format. Secondly, the information classification system is the other issue. Without using the appropriate classification system, people find that delivered information is often redundant. Therefore, an automated classification system that selects only the relevant information for each user is required. The main research task for this system is how to implement the incremental knowledge acquisition process for the classification knowledge because human classification knowledge is always heuristic and changes rapidly and, therefore, it is necessary to maintain the knowledge base incrementally.

5. Personnel

Principle Investigators

Dr. Byoeng Ho Kang
University of Tasmania

Co-Principle Investigators

Prof. Paul Compton
University of Tasmania

Prof. Hiroshi Motoda
University of Tasmania

Research Students

Yang Sok Kim
Ph.D Student, School of Computing, University of Tasmania

Sung Sik Park
Ph.D Student, School of Computing, University of Tasmania

6. Publications: Papers are attached in Appendix

Gil Cheol Park, Seok Soo Kim, Gun Tae Bae, Yang Sok Kim and Byeong Ho Kang (2006). An Automated WSDL Generation and Enhanced SOAP Message Processing System for Mobile Web Services. Third International Conference on Information Technology: New Generations (ITNG 2006), Las Vegas, Nevada, USA.

Yang Sok Kim, Young Ju Choi, Sung Sik Park, Gil Cheol Park, Seok Soo Kim and Byeong Ho Kang (2006) Knowledge Acquisition in Open-ended Document Classification Problem. The 19th Australian Joint Conference on Artificial Intelligence, Hobart, Australia (Under Review)

7. Interactions

Research collaboration meeting with the meta-search engine research team under Dr. John Salerno in Air Force Research Lab in Rome.

Sample demo Web site development meeting with Dr. Park, Tae Woo from AOARD and email communications to survey the requirements from the field.

Appendix 1. The paper published in ITNG

Gil Cheol Park, Seok Soo Kim, Gun Tae Bae, Yang Sok Kim and Byeong Ho Kang (2006). An Automated WSDL Generation and Enhanced SOAP Message Processing System for Mobile Web Services. Third International Conference on Information Technology: New Generations (ITNG 2006), Las Vegas, Nevada, USA.

An Automated WSDL Generation and Enhanced SOAP Message Processing System for Mobile Web Services*

Gil Cheol Park[1], Seok Soo Kim[1], Gun Tae Bae[1], Yang Sok Kim[2] and Byeong Ho Kang[2]
[1]*School of Information & Multimedia, Hannam University*
133 Ojung-Dong, Daeduk-Gu, Daejeon 306-791, Korea
gcpark@mail.hannam.ac.kr
[2]*School of Computing, University of Tasmania*
Sandy Bay, Tasmania 7001, Australia
{yangsokk, bhkang}@utas.edu.au

Abstract

Web services are key applications in business-to-business, business-to-customer, and enterprise applications integration solutions. As the mobile Internet becomes one of the main methods for information delivery, mobile Web Services are regarded as a critical aspect of e-business architecture. In this paper, we proposed a mobile Web Services middleware that converts conventional Internet services into mobile Web services. We implemented a WSDL (Web Service Description Language) builder that converts HTML/XML into WSDL and a SAOP (Simple Object Access Protocol) message processor. The former minimizes the overhead cost of rebuilding mobile Web Services and enables seamless services between wired and wireless Internet services. The latter enhances SOAP processing performance by eliminating the Servlet container (Tomcat), a required component of typical Web services implementation. Our system can completely support standard Web Services protocol, minimizing communication overhead, message processing time, and server overload. Finally we compare our empirical results with those of typical Web Services.

1. Introduction

As the Internet potentials of the mobile Internet are widely understood, mobile Internet services become a major mediator in information delivery and in business transactions. Mobile Internet services, however, still have physical devices, network and content limitation. Firstly, mobile devices are limited by system resources such as smaller screens and less convenient input devices. Secondly, wireless networks have less bandwidth, less connection stability, less predictability and a lack of standardized and higher costs [1, 2]. Lastly, mobile Internet services also have content limitation because the amounts of available mobile content are still smaller than that of wired Internet services, and the consistency between wired and wireless Internet services is very critical. Physical device and network limitation make supporting common Internet standards, such as HTML, HTTP, and TCP/IP, difficult because they are inefficient over mobile networks. Therefore, new protocols such as WAP (Wireless Application Protocol) and WML (Wireless Markup Language) are proposed to overcome these limitations. Content limitation encourages researchers to find methods that support reuse of current wired Web information. Some researchers focus on the conversion of HTML documents to mobile Internet serviceable WML documents and direct access to databases, to provide efficient information delivery in the wireless environment [3-7]. However, these researchers do not focus on the capability that allows applications to interact over the Internet in an open and flexible way, but on the capability that provides dynamic wireless Internet service according to different network and device environments. The former goal can be achieved by Web Services, because interactions between Web Services applications are expected to be independent from the platform, programming language, middleware, and applications involved. For this reason, Web Services is regarded as key applications in business-to-business, business-to-customer, and enterprise applications integration solutions [8].

In this paper, we focus on the following two issues:

Automated HTML/XML conversion to WSDL: The goal system should dynamically generate WSDL files from existing HTML/XML files. A markup language converting system is implemented to convert HTML/XML to WSDL automatically.

Improve SOAP processing efficiency: One main limitation of Web Service is its inefficient performance

* This work is supported by the Asian Office of Aerospace Research and Development (AOARD) (Contract Number: FA5209-05-P-0253)

compared with other distributed computing approaches like Java RMI, CORBA, and DCOM (Distributed Component Object Model). The use of HTTP and XML represents a significant increase in run-time cost Web Services solutions [9-13]. We propose a method that enhances SOAP processing by changing service architecture. The typical SOAP processing system requires the Web Servlet container (e.g. Tomcat) to execute SOAP. It requires additional process and communication port. Our hypothesis is that if a system processes the SOAP message directly, without help from Web Servlet container, the SOAP performance improves.

The paper is organized as follows: Section 2 summarizes relevant research results, including HTML conversion and Web services technology. Section 3 explains our HTML conversion implementation, while Section 4 illustrates our SOAProc system implementation. In Section 5 we compare our system's performance with the typical Web Services implementation approach. Finally, conclusions and recommendations for further work are described in Section 6.

2. Literature Review

Researchers usually focus on HTML/WML conversion because the WAP is an alternative protocol for HTML in wireless Internet services using Wireless Markup Language (WML), a small subset of Extensible Markup Language (XML), to create and deliver content. Kaasinen et al. [3] and Dugas [14] suggested an HTML/WML conversion proxy server, which converts HTML-based Web content automatically, and on-line, to WML. Saha et al. [6] suggested a middleware that is seamless and transparently translates a Web site's existing contents to mobile devices. Kurbel and Dabkowski [4] proposed a dynamic user tailed WML content generation by using JSP (Java Server Pages) and JDBC-ODBC driver. Magnusson and Stenmark [15] suggested a CMS-based approach to visualise Web information in a PDA. Pashtan et al. [7] stressed context-aware wireless Web services, which can adapt their content to the user's dynamic content. Again, we wish to stress these researchers focus only on HTML/WML conversion, not Web Services compliable conversion. Therefore, in spite of their importance, application integration aspects inside and outside enterprises have not been seriously considered by the researchers. As the impotence of application integration over the Internet becomes more important, nowadays Web Services are critical to any Internet services. For this reason, we propose a method that converts HTML to WSDL. The WSDL files are used to provide Web Services with SOAP messaging protocols. More detailed explanation about the Web Services and its implementation issues are discussed in the following Section.

Web Services, as defined by the W3C Web Services Architecture Working Group, are "software applications identified by a URI, whose interfaces and bindings are capable of being defined, described, and discovered as XML artefacts. A Web service supports direct interactions with other software agents using XML-based messages exchanged via Internet-based protocols."[16]

There are several Web Services implementation methods, which differ in their support for class binding, ease of use, and performance [13]. Among them Apache Axis (Apache eXtensible Interaction System) with Tomcat is a popular implementation method. The Apache Axis project is a follow-on to the Apache SOAP project and currently has reached version 1.2.1, but it's not part of the Apache SOAP project. Axis is a completely new rewrite with emphasis on flexibility and performance. It supports HTTP SOAP request/response generation, SOAP message monitoring, dynamic invocation, Web service deployment, and automatic WSDL generation for Web services.

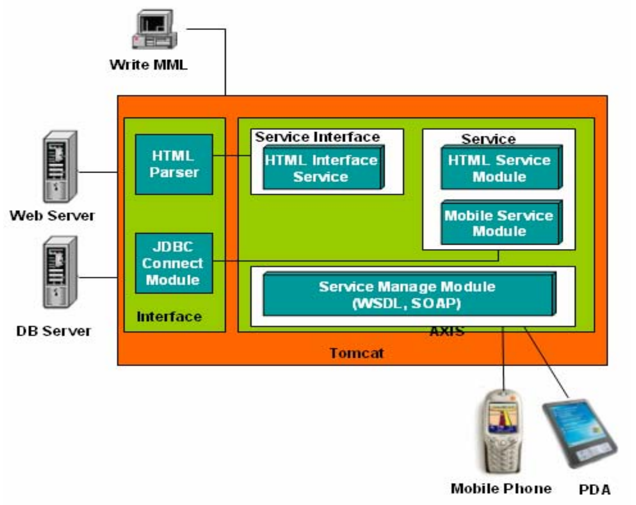


Figure 1- Typical Mobile Web Service Implementation with AXIS and Tomcat

Figure 1 illustrates this standard mobile Web service implementation. A Web Servlet container, like Tomcat, is required to provide mobile Web services with Axis. For wireless Internet service, the server administrator should write MML (Made Markup Language) to parse Web contents by using the administrative tool. A MML is used to generate service request forms or service results by dynamically parsing the existing Web contents and sending them to relevant model and clients. When a client, whether it is wireless or wired client, requests Web service via SOAP request, Apache Tomcat transfers it to Axis. Axis interfaces the SOAP request message into a relevant service by using the service management function. Service providing models are interfaced by using a WSDL module is provided by Axis. By implementing SOAP and distributed computing service, the system architecture can have a lightweight thin client structure and the service can be provided in a flexible way. However, this implementation is not efficient because it requires additional process for Web Servlet engine (Tomcat) and communication port. For this reason, we propose an alternative system that can process SOAP messages without using Web Servlet engine.

3. HTML/WSDL converter

Our HTML/WSDL content converter system consists of three sub-modules: the rule script, the script engine, and the markup language converter. The rule script stores rules for content reformatting rules, which are created by the user with the management program. The rules include personalization information and display structuring information of mobile devices. Secondly, the script engine reconstructs contents by using script rules and client (device) information. The markup language converter transforms markup language if the markup language that the server provides differs from what the client can process. Script rules are created as follows. If a Web site address is supplied, our system reads and parses the Web site information. The parsed information is then presented by using a DOM tree, in which the user can select and save node information to be served as wireless Internet content. The JML (Java Mark-up Language) editor defines XML tags and attributes of the saved items. TITLE, BASEURL, LINK, HREF, CONTENT, and ELEMENT are XML tag examples and many attributes are also available to customize mobile contents.

Figure 2 illustrates the operation of the converter. The user accesses the HTML/WSDL converter system via mobile devices and mobile networks. The converter gets the user's mobile device information such as display size and color, and URL information that the user requests by using the protocol detector. After getting this information, the converter requests URL information from the Web server. The Web server generates a HTML response message and sends it to protocol detector. The protocol detector then passes this HTTP response message to the selector with client information. The selector chooses WSDL information from the HTTP response message by using the script rules and returns this information to the protocol detector. The protocol detector in turn sends this information to the translator, which performs Mark-up language transformation, image transformation, paging and cashing. Lastly, the converter sends this processed result to the user.

4. SOAP Message Processor

In the Web Services, XML based SOAP messages are used when the clients request Web Services from the server or when the server sends Web Service response messages to the clients.

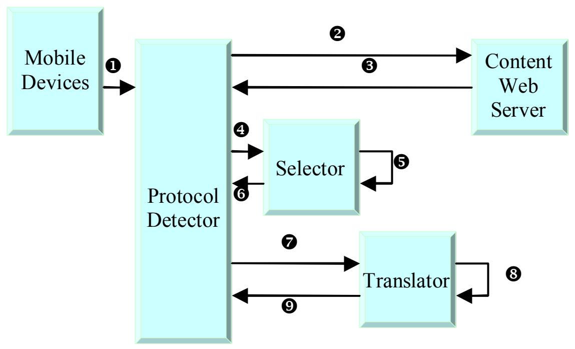


Figure 2- HTML/WSDL Content Converter Operation

In the standard Web Services implementation this is supported by Tomcat and AXIS. We developed a SOAP message processing system, called SOAProc, because typical architecture causes inefficiency by spawning new process and adding additional communication port. The SOAProc directly processes the SOAP request and response messages without using Servlet engine. Figure 3 illustrates our Web Services system implementation architecture, in which the SOAProc and the WSDL builder are used. The most significant difference between the standard system (see Figure 1) and our implementation (see Figure 3) is that our system does not include Tomcat. Instead of using Tomcat's WSDL and SOAP supporting function, WSDL files are directly generated by the WSDL builder and SOAP messages are processed by the SOAProc system.

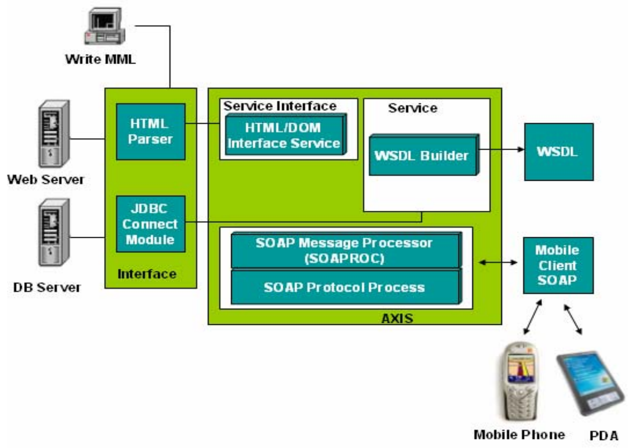


Figure 3 – Mobile Web Service using the SOAProc and the WSDL builder

4.1 SOAP Message Structure

Figure 4 illustrates an example of a SOAP message. The Header element is intentionally omitted in this example. <ns1: IntranetLogin …> indicates IntranetLogin method that will be called. The tags between

<ns1:IntranetLogin…> tag are parameters of method IntranetLogin, such as <userid> … </userid>, <pass> … </pass>, and <sessionidtag> … </sessionidtag>.

```
RequestSoapMassage.xml
```

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv =
"http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance">
<soapenv:Body>
    <ns1:IntranetLogin soapenv:encodingStyle
    =http://schemas.xmlsoap.org/soap/encoding/"
    xmlns:ns1="urn:wireserver">
        <useridxsi:type="xsd:string">
            test
        </userid>
        <pass xsi:type = "xsd:string">
            pass
        </pass>
        <sessionidtag xsi:type="xsd:string">
            sessionidtag
        </sessionidtag>
    </ns1:IntranetLogin >
</soapenv:Body>
</soapenv:Envelope>
```

Figure 4 - SOAP Message Example

4.2 SOAP Request Message Analysis

The algorithm that is used to analyses the method and its parameters of SOAP request messages is as follows:

Step1: Gets the SOAP messages

The system generates a FileInputStream of RequestSoapMessage.xml (fis).

```
FileInputStream fis = new
FileInputStream("RequestSoapMassage.xml")
SOAPEnvelope env = new SOAPEnvelope(fis);
```

Then the system gets the SOAP message from the above FileInputStream.

```
SOAPBodyElement sbe = env.getFirstBody();
```

Step2: Gets the SOAP Body

The system extracts the SOAP Body from the SOAP message.

```
sbe.getName();
```

Step3: Analysing SOAP Body

The system finds IntranetLogin part of <ns:IntranetLogin …> from the Body of the SOAP message.

```
ArrayList al = sbe.getChildren();
```

The system creates array list of items between <ns:IntranetLogin…></ ns:IntranetLogin> in the Body of the SOAP message.

```
for(int i = 0 ; i < al.size() ; i++){
   MessageElement me = MessageElement)(al.get(i));
   System.out.println((i+1)+" th " +
   me.getName()+"'s value is " + me.getValue());
}
```

The system iteratively analyses the item list to get MessageElement like <userid> … </userid>, <pass> … </pass>, and <sessionidtag> … </sessionidtag>. In each iteration, the item's name and value are obtained by me.getName() and me.getValue()method. For example, if the system uses example in Figure 6, <userid … >test </userid> is in the first item of item list and 'userid' and 'test' are name and value, which can be get by using iterative analysis. The system can generate response message to the clients by using this result.

4.3 SOAP Response Message Generation

Our system analyses the client SOAP request message and sends the analyzing result to the Web server. When the Web server system generates a HTTP response message, our system generates a SOAP response message by using it. In this part, we describe a response generation algorithm, in which we assume the response result is a string type. The response results can be sent by a single or binary array. The result values and method namespace value are assumed as follows.

```
String str = "test1Respons";
String strElement = "test1Return";
String elementValue = "aaa";
String nameSpaceURI = "urn:stringtest";
```

The SOAP response message is generated as follows:

Step 1: Generate SOAP Basic Element

Our system generates the new SOAP response message by creating a new Envelope element and Body element.

```
SOAPEnvelope env = new SOAPEnvelope();
env.getBody(); //SOAPBody
SOAPBodyElement body = new SOAPBodyElement();
```

The SOAP message that is created until now is as follows:

```
<soapenv:Envelope
xmlns:soapenv=http://schemas.xmlsoap.org/soap/enve
lope/
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance">
<soapenv:Body />
</soapenv:Envelope>
```

Step 2: Add Content to SOAP Message

The SOAP contents are created by adding the above results values.

```
RPCParam rpcParam = new RPCParam(strElement,
elementValue) ;
RPCHeaderParam rpcHeaderParam = new
RPCHeaderParam(rpcParam);
RPCElement rpcElement = new RPCElement(str);
rpcElement.addParam(rpcParam);
rpcElement.setEncodingStyle("http://schemas.xmlsoa
p.org/soap/encoding/");
  rpcElement.setNamespaceURI(nameSpaceURI);
</soapenv:Envelope>
```

After adding the contents, the SOAP response message is as follows:

```
<ns1:test1Respons
soapenv:encodingStyle="http://schemas.xmlsoap.org/
soap/encoding/"
xmlns:ns1="urn:stringtest">
<test1Return
si:type="xsd:string">aaa</test1Return>
</ns1:test1Respons>
```

Step 3: Error Handling

If there are any errors in the Web server processing, the following code creates error messages.

```
SOAPFault soapFault = env.getBody().addFault();
soapFault.setFaultCode("code error\n");
soapFault.setFaultActor("action error\n");
soapFault.setFaultString("string error\n");
```

Step 4: Add SOAP Body Element

Lastly, the following code adds the SOAP Body element when there is no error. SOAP response generation is completed by doing this.

```
env.addBodyElement(rpcElement);
```

Figure 5 illustrates a complete SOAP response message that is generated by our system without Tomcat.

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/env
elope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance">
<soapenv:Body>
   <ns1:test1Respons
soapenv:encodingStyle="http://schemas.xmlsoap.org/
soap/encoding/" xmlns:ns1="urn:stringtest">
   <test1Returnxsi:type="xsd:string">
         aaa
   </test1Return>
   </ns1:test1Respons>
</soapenv:Body>
</soapenv:Envelope>
```

Figure 5 - SOAP Response Message

5. Experiment

5.1 Method

The experiment is focused on the performance evaluation of our mobile Web service system. Two sets of systems are prepared for our experiment. The first system is implemented with standard Web Services architecture as explained in Section 2. This implementation requires Tomcat Servlet container with AXIS. The second implementation is based on our approach. Where there is no Servelet container with the SOAP message processing performed by the SOAProc system and WSDL created by the WSDL builder. We conducted a simulated performance comparison experiment. Figure 6 illustrates the experiment process. If a client requests Web services by submitting a SOAP request, the experiment system analyses the SOAP message and sends a HTTP request to the content Web servers. If the experiment system receives a HTTP response message form the Web server, it generates WSDL and sends a SOAP response message to the clients' mobile device.

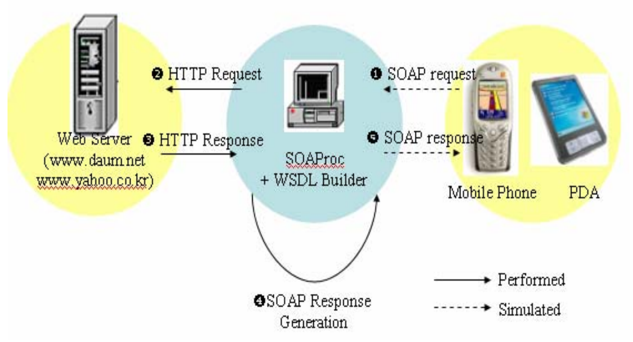


Figure 6 – Experiment System Procedure

SOAP requests are simulated by the mobile client simulation program, which connects to the experiment system and sends several SOAP request messages. There are time intervals, from 1 to 10 seconds between SOAP requests. If the connection is closed, the simulation program continually tries to connect to the experiment system. We assumed that there were 200 users at the same time. SOAP requests were created by four client programs and each program generated 50 threads at the same time. We chose two public Web sites [www.daum.net (dictionary) and www.yahoo.co.kr (stock)], which role as the content Web servers in our experiment. We assumed two kinds of specific information - dictionary and stock - are required by the user from these Web servers. Each service's timeout is 30 seconds.

The following results were collected to compare two experiment systems:

- Test time: how many seconds were consumed for the test.
- Number of Requests: how many requests were generated within test time.
- Connection Timeout: the connection numbers that were not connected within the request timeout.
- Connection Refuse: the request numbers that could not be connected because the server was busy.
- Connection Handshake Error: the number of session configuration failures after connection
- Connection Trial Time: How many times the client could not connect to the server.

- Request Timeout: how many times the timeout was exceeded.

5.2 Results

Table 1 summarizes the experiment as results, which illustrate an enhanced performance in all categories. Though the test time of our system is shorter than that of the standard system, the total number of requests is greater than that of standard system and the timeout number is less than that of the standard system. For example, whist the average request per second of our system is 17.94, that of standard system is 9.64. There are many connection errors in the standard system. Only some portion of 200 requests is successfully connected to the server while the others get a "refused" message from the server. However, those kinds of connection failures do not happen in our system.

Table 1 – Experiment Results

	SOAProc System	Standard System
Test time	29,400	46,200
Total Request	524,573	445,422
Connection Timeout	0	435
Connection Refused	0	18,960
Connection Handshake Error	0	513
Connection Trials	243	22,534
Request Timeout	756	119,891

6. Conclusions

Mobile Web services are critical solutions in the Internet service integration architecture. In this research we proposed a new Web Service architecture by implementing two significant systems. Firstly, the HTML/WSDL converter can support reusing current HTML based contents. This is essential for saving developing or maintenance costs and serving seamless Internet services both wired and wireless. Secondly, we proposed a new SOAP message processing system to diminish SOAP latency problems by eliminating the Tomcat Servelet container in the Web Services implementation. The SOAP request and response messages are directly processed by the SOAProc system. We can implement an alternative mobile Web Services system by using these two systems without violating standard Web Services protocols. Our system can process more service request about doubly efficient than that of typical Web service implantation with very small connection errors.

7. References

1. Siau, K., E.P. Lim, and Z. Shen, *Mobile commerce: promises, challenges, and research agenda.* Journal of Database Management, 2001. **vol.12, no.3**: p. 4-13.
2. Kim, H., et al. *An Empirical Study of the Use Contexts and Usability Problems in Mobile Internet*. in *35th Annual Hawaii International Conference on System Sciences (HICSS'02)*. 2002.
3. Kaasinen, E., et al., *Two approaches to bringing Internet services to WAP devices.* Computer Networks, 2000. **33**(1-6): p. 231-246.
4. Kurbel, K. and A. Dabkowski. *Dynamic WAP content Generation with the use of Java Server Pages*. in *Web Databases/Java and Databases: Persistence Options (Web&DB/JaDa)*. 2002. Erfurt, Germany.
5. Metter, M. and R. Colomb. *WAP Enabling Existing HTML Applications*. in *First Australasian User Interface Conference*. 2000.
6. Saha, S., M. Jamtgaard, and J. Villasenor, *Bringing the wireless Internet to mobile devices.* Computer, 2001. **vol.34, no.6**: p. 54-58.
7. Pashtan, A., S. Kollipara, and M. Pearce, *Adapting content for wireless Web services.* IEEE Internet Computing, 2003. **7**(5): p. 79-85.
8. Farrell, J.A. and H. Kreger, *Web services management approaches.* IBM Systems Journal, 2002. **vol.41, no.2**: p. 212-227.
9. Seshasayee, B., K. Schwan, and P. Widener, *SOAP-binQ: high-performance SOAP with continuous quality management.* Proceedings. The 2nd IEEE International Conference on Distributed Computing Systems, 2004: p. 158-165.
10. Kohlhoff, C. and R. Steele, *Evaluating SOAP for high performance applications in capital markets.* Computer Systems Science and Engineering, 2004. **19**(4): p. 241-251.
11. Chiu, K., M. Govindaraju, and R. Bramley, *SOAP for High Performance Computing*, in *11 th IEEE International Symposium on High Performance Distributed Computing HPDC-11 20002 (HPDC'02)*. 2002, Indiana University. p. 246.
12. Chiu, K., M. Govindaraju, and R. Bramley. *Investigating the Limits of SOAP Performance for Scientific Computing.* in *11th IEEE International Symposium on High Performance Distributed Computing (HPDC-11 '02)*. 2002.
13. Davis, D. and M. Parashar. *Latency Performance of SOAP Implementations.* in *2nd IEEE/ACM International Symposium on Cluster Computing and the Grid*. 2002.
14. Dugas, R., *WWW Unplugged: An HTML to WML transcoding proxy.* 2001.
15. Magnusson, M. and D. Stenmark. *Mobile Access to the Intranet: Web Content Management for PDAs.* in *Americas Conference on Information Systems 2003*. 2003.
16. W3C, *Web Services Architecture Requirements.* Web Services Architecture Requirements, 2002.

Appendix 2. The paper under review in AI 2006

Yang Sok Kim, Young Ju Choi, Sung Sik Park, Gil Cheol Park, Seok Soo Kim and Byeong Ho Kang (2006) Knowledge Acquisition in Open-ended Document Classification Problem. The 19th Australian Joint Conference on Artificial Intelligence, Hobart, Australia (Under Review)

Knowledge Acquisition in Open-ended Document Classification Problem[†]

Yang Sok Kim[1], Young Ju Choi[1], SungSik Park[1], Gil Cheol Park[2], and Seok Soo Kim[2]

[1]School of Computing, University of Tasmania
Sandy Bay, Tasmania 7001, Australia
{yangsokk, Y.J.Choi,sspark}@utas.edu.au

[2]School of Information & Multimedia, Hannam University
133 Ojung-Dong, Daeduk-Gu, Daejeon 306-791, Korea
{sskim, gcpark}@hannam.ac.kr

Abstract. This study focused on the open-ended problem, which differs from the close-ended one. Open-ended problem solving is characterized by the active role of problem solvers, multiple solutions for the same problem, and the negotiating interactions between the problem solver and the learner. We employed an open-ended problem solving approach to solve a real-world document classification problem, which is an example of the open-ended problem. Our approach is implemented by MCRDR, an incremental knowledge acquisition method. We conducted knowledge acquisition experiments with the MCRDR based document classification system, called MCRDR-Classifier. Twenty participants classified 12,784 articles collected from eight IT news Websites by using the MCRDR-Classifier for four months. Our experiment results show that our approach is appropriate for the open-ended document classification problem.

1 Introduction

Knowledge acquisition is a core part of any knowledge based system; it transfers human knowledge into the system and enables the knowledge based system to perform human-like operations. Whenever knowledge acquisition is conducted, whether it is conducted by either knowledge engineering (KE) approach or machine learning (ML) approach, it is clear that the task environments and problem spaces are quietly different among domains. Therefore, it is natural to search for an appropriate classification system and class of problems in order to find the optimal solution for those problems. One of the best known problem classifications is to divide problems into the close-ended and the open-ended categories. Sometimes these are known as well-structured and ill-structured problems, because the open-ended problems are typically ill-defined

[†] This work is supported by the Asian Office of Aerospace Research and Development (AOARD) (Contract Number:FA5209-05-P-0253)

whereas the close-ended problems are well-defined [1, 2]. These two types of problems have quite different characteristics. A closed domain would be a subject area like arithmetic, where there are often patently specified problem solving goals, correct answers and clearly defined criteria for success at problem solving. An open-ended domain, like writing, is where there is typically a lack of clearly defined goals, no single correct solution to a problem and no immediately obvious criteria for making a judgment as to what constitutes a correct solution [3, 4]. The close-ended problem solving research has been made considerable progress and significantly contributed to the both research and commercial community. However, it is beneficial to look into open-ended problem solving approach because most people share the intuition that there are important differences in these two types of problem solving approach. It is not *a priori* clear that the results from the close-ended research will generalize the open-ended domains [3].

In this research, we studied knowledge acquisition in an open-ended document classification problem. Though there is recent research in the open-ended problem spaces, the traditional document classification is a close-ended problem. The goal of document classification is to classify unseen cases efficiently into pre-defined categories (classes) by using specific algorithms. Data (cases) used for the study are pre-processed and usually have assigned class or classes. Some parts of these data are used to train classifiers and the other parts are used to test the classifiers performance. The classes that the cases are classified into are also pre-defined. The performance of each classifier is measured by the well developed evaluation metrics such as precision and recall. However, the real-world document classification is an open-ended problem though its goal is very similar to that associated with traditional classifications. Cases that should be classified are not known until they are presented and classes are not pre-defined and continually evolve as time passes. Some parts of classes newly appear in the problem space and the other parts of classes are depreciated. In addition, it is very difficult to use traditional performance metrics to decide the success of classifiers. For these reasons, we needed to employ a method that differs from the traditional document classification approach. To this end, we explored the open-ended education experiences because the open-ended problem-solving approaches have been employed in the education area for a long time[5]. We employed the MCRDR (Multiple Classification Ripple-Down Rules) method as an implementing method for the open-ended document classification system.

This paper consists of the following contents. We used some insights from the open-ended education because the open-ended approaches were tried in the education as an alternative of traditional education. Their contribution to our study is discussed in the Section 2. The background information about the MCRDR method is described in Section 3 and implementation details of our document classifier, called MCRDR-Classifier, is in Section 4. Inference process and knowledge acquisition in the MCRDR-Classifier are discussed in Section 5 and Section 6. We conducted real-world document classification experiments with our classifier and analyzed knowledge acquisition behaviors in the open-ended domains. Section 7 explains our experiment design and Section 8 summarizes experiment results. Conclusions of this paper are described in the Section 9.

2 Insights from the Open-Ended Education Experience

Open-ended problem solving approaches have been studied in the education area as an alternative to the traditional education method. It was beneficial for us to examine their experience to get some insights into problem solving of knowledge acquisition problem. According to the open-ended educationist, in the open-ended domains like music composition or writing, the learning goal is not fixed, but may change during the course of the negotiation process. The open-ended domain usually requires some form of open interactions between teacher and learner. This shifts the emphasis away from the assertion of facts and towards interactions that encourage the type of creative, meta-cognitive and critical thinking [4, 5]. In this research, we focused on three insights listed below and examine why they are also very important in the open-ended knowledge acquisition.

Insight (a) Active Role of Problem Solver: The open-ended approach in education emphasizes the active role of the teacher (problem-solver) in the problem process [5]. There are two different approaches in the knowledge acquisition: the constructivist approach vs. the rational/objectivist approach. This latter approach defines a problem as the observable gap between a given goal, or predefined standards of performance, and the present state of affairs ignoring the presence of the problem-solver. At the very least, it reduces the problem-solver to a container for replicable problem-solving procedures. The former approach reflects a view which recognizes the problem-solver as central to problem solving and knowledge is only used as it is seen through the eyes of the problem-solver. [6, 7] The open-ended education approach is philosophically similar to that of the constructivist approach in that both of them emphasize the active role of the problem solver. Therefore, the promising KA tools should help the problem solvers to perform easy KA. We did not employ the machine based approaches, because they basically remove the problem solver from the problem solving process, moreover the problem solver cannot directly affect the problem solving process. Instead, we employed a rule-based knowledge engineering approach and made the problem solvers directly interact with the KA tool to encourage the problem solver to perform knowledge acquisition more actively.

Insight (b) Multiple Solutions for the Same Problem: The learning goal of open-ended education is not fixed, but may change according to context. The problem solvers may recognize the same context in different ways because there are individual differences in human cognition such as short-term and long-term memory, verbal processes, and solution strategy [8]. Moreover, each problem solver can provide different solutions even though they interpreted the context in the same way. The constructivist knowledge acquisition is based on similar assumptions. It assumes that "*the goal may be shared with others but there may be numerous individual methods of* *reaching the goal.*"[6] In the classification problem, this has a two-fold meaning. On one hand, it means that ***(1) a case can be classified into multiple classes*** without making dissatisfactions among the problem solvers. On the other hand, ***(2) multiple cases can be classified into one class*** because of different reasons.

Insight (c) Importance of negotiating interaction: The negotiating interactions between the problem solver and the learner are very important in open-ended education. The negotiations are related to the level and/or amount of knowledge that is transferred from the problem solver to the learner. The problem solver only provides additional knowledge when the learner can understand or use that knowledge sufficiently. In the course of education, these understanding are incremental and interactive process. As the problem solver knows more about the domain and the learners, he/she can transfer more knowledge to the learner [1, 4, 5, 9]. We can view the knowledge based system as the learner in the KA process because the outputs from the system are continually evaluated by the problem solvers. This evaluation processes continues until the problem solver completely satisfies the system's suggestions. In addition, the problem solver usually increases his/her understanding about the domain incrementally while he/she transfers his/her knowledge to the system. Therefore, as the problem solver in an education environment provides his/her knowledge to the learner incrementally, the problem solver in the knowledge acquisition environment usually transfers his/her knowledge into the system gradually to manage current salient cases. From this point of view, the knowledge acquisition is not a once-for-all process, but a continual process, and knowledge is continually patched by new knowledge. Therefore, the KA system should support this kind of patching work effectively [7].

3 MCRDR

We employed MCRDR (Multiple Classification Ripple-Down Rules) to implement an open-ended KA tool for the document classification system. It is proposed to overcome the knowledge acquisition problem based on a maintenance experience of a real world medical expert system called GARVAN-ES1 [10, 11]. The MCRDR method is based on constructivism and focuses on the problem solvers emphasizing their direct problem solving, not the indirect problem solving by the knowledge engineer.[12] Knowledge in the MCRDR systems is regarded as not permanent but temporary, which means current knowledge is true only if the new situation is consistent with the old one. The problem solver interacts with the KA system improving its problem solving capability incrementally. Therefore, maintenance of the knowledge base is a core process of the KBS development, not the additional process or aftermath of the development. [11] For these reasons, easy knowledge base (KB) maintenance is a key goal of the MCRDR based system.

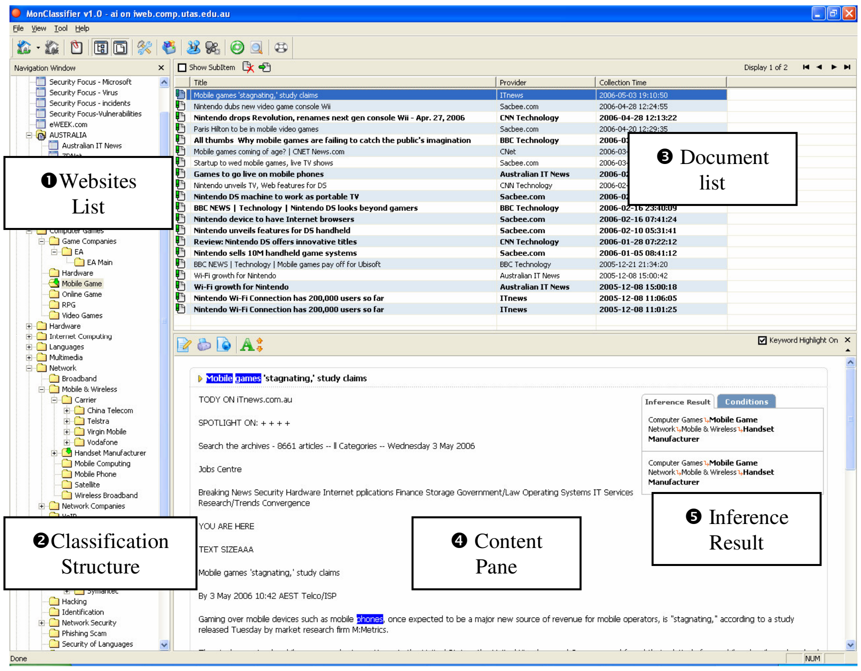
The MCRDR KB is an n-ary tree structure. MCRDR uses a "***rules-with-exceptions***" knowledge representation scheme because the context in the MCRDR is defined as the sequence of rules that were evaluated as leading to a wrong conclusion, or no conclusion, with the existing knowledge base [13]. This approach makes maintenance easier than that of the traditional approach because the maintenance process only takes place in relation to current rule and its children rules. A classification recommendation (conclusion) is provided by the last rule satisfied in a pathway. All children of the satisfied parent rule are evaluated, allowing for multiple conclusions. The conclusion of the parent rule is only given if none of the children are satisfied [13-15]. There are two types of rule. The refining rule is placed under the root rule or other rules to refine current rule and the stopping rule is placed under the root rule or other rules to stop current rule. Each rule is created when the problem solver dissatisfies current inference result and wants to make an exception for the current rule. The cases used for the rule creation are called "**cornerstone cases**" and are saved while the new rule is created. The cornerstone cases help the problem solver by showing cases instead of abstracted rules, because the classification context can be easily retrieved with the cases. In addition to the ***cornerstone case***, MCRDR uses a ***difference list*** to support easy KB maintenance. The difference list is created by the comparing current case's attributes with those of the relevant rule's cornerstone cases. The scope of the relevant rule is the current firing rule and its children rules. The MCRDR systems make even naïve domain experts maintain a very complicated knowledge base without the knowledge engineer's help [7, 12]. A prior study shows that the MCRDR method guarantees low cost knowledge maintenance in the real world domain [13, 16].

4 MCRDR Document Classification System (MCRDR-Classifier)

A document classification system is implemented with the MCRDR algorithm (MCRDR-Classifier), C++ program language, and MySQL database. A Website monitoring system, called WebMon, continually provides newly updated documents from the targeted Websites. The WebMon detects new information, pre-processes it to extract core content from the Webpage source, and provides it to the MCRDR-Classifier. Detailed explanations are described in [17, 18].

Fig. 2. illustrates the MCRDR-Classifier's main user interface. The left pane shows the tracking Websites (❶) that are selected by the problem solver and the classification structure of the problem solver (❷). The problem solver can choose his/her favorite tracking Websites that are maintained by WebMon. When the problem solver selects one Website in the list, the newly collected documents that are ordered by collected time are displayed in the right upper pane (❸). When one document is selected, the content of the document is displayed in the content pane (❹) with a detailed explanation of inference results (❺) and the inference results in ❷ by displaying downward arrow (s). The problem solver can see the documents that are classified into the specific class (folder) by selecting the folder from❷. The conditions of fired rule sets are also displayed and by selecting the condition tap in the explanation box (❺) and they are also highlighted in the content pane (❹). More detailed inference procedures are explained in the following section.

One important thing is that the MCRDR classifier supports multiple classifications. For example, there are two destine folders (Mobile game and Handset Manufacturer) for the document “Mobile games 'stagnating,' study claims”. In this case two independent rules are used to classify this document into two different folders. In the other case, one document can be classified into one destine folder by multiple rules. The figure that displays this case is not included in this paper due to the lack of space.



Title	Provider	Collection Time
Mobile games 'stagnating,' study claims	ITnews	2006-05-03 19:10:50
Nintendo dubs new video game console Wii	Sacbee.com	2006-04-28 12:24:55
Nintendo drops Revolution, renames next gen console Wii - Apr. 27, 2006	**CNN Technology**	**2006-04-28 12:13:22**
Paris Hilton to be in mobile video games	Sacbee.com	2006-04-20 12:29:35
All thumbs Why mobile games are failing to catch the public's imagination	**BBC Technology**	**2006-0**
Mobile games coming of age? | CNET News.com	CNet	2006-03
Startup to wed mobile games, live TV shows	Sacbee.com	2006-03
Games to go live on mobile phones	**Australian IT News**	**2006-0**
Nintendo unveils TV, Web features for DS	CNN Technology	2006-02
Nintendo DS machine to work as portable TV	**Sacbee.com**	**2006-0**
BBC NEWS | Technology | Nintendo DS looks beyond gamers	**BBC Technology**	**2006-02-16 23:40:09**
Nintendo device to have Internet browsers	**Sacbee.com**	**2006-02-16 07:41:24**
Nintendo unveils features for DS handheld	**Sacbee.com**	**2006-02-10 05:31:41**
Review: Nintendo DS offers innovative titles	**CNN Technology**	**2006-01-28 07:22:12**
Nintendo sells 10M handheld game systems	**Sacbee.com**	**2006-01-05 08:41:12**
BBC NEWS | Technology | Mobile games pay off for Ubisoft	BBC Technology	2005-12-21 21:34:20
Wi-Fi growth for Nintendo	Australian IT News	2005-12-08 15:00:42
Wi-Fi growth for Nintendo	**Australian IT News**	**2005-12-08 15:00:18**
Nintendo Wi-Fi Connection has 200,000 users so far	**ITnews**	**2005-12-08 11:06:05**
Nintendo Wi-Fi Connection has 200,000 users so far	**ITnews**	**2005-12-08 11:01:25**

Mobile games 'stagnating,' study claims

TODY ON iTnews.com.au

SPOTLIGHT ON: + + + +

Search the archives - 8661 articles -- ll Categories -- Wednesday 3 May 2006

Jobs Centre

Breaking News Security Hardware Internet pplications Finance Storage Government/Law Operating Systems IT Services Research/Trends Convergence

YOU ARE HERE

TEXT SIZEAAA

Mobile games 'stagnating,' study claims

By 3 May 2006 10:42 AEST Telco/ISP

Gaming over mobile devices such as mobile phones, once expected to be a major new source of revenue for mobile operators, is "stagnating," according to a study released Tuesday by market research firm M:Metrics.

Computer Games↳**Mobile Game**
Network↳Mobile & Wireless↳**Handset Manufacturer**

Computer Games↳**Mobile Game**
Network↳Mobile & Wireless↳**Handset Manufacturer**

Fig. 2. Main Interface of the MCRDR-Classifier

5 Inference Process of the MCRDR- Classifier

Basically, the inference process implementation of the MCRDR-Classifier follows the MCRDR method that is described in [19]. The WebMon continually revisits target Websites and report newly found hyperlinks as new information. The hyperlink text is used as “Title” in the MCRDR-Classifier because though it may or may not match with the title of the hyperlinked document, it is usually employed to represent the hyperlinked document. The hyperlinked document contains not only the main content but also noisy or redundant content such as advertisements, navigation or decorative content. Though we can use other features such as hyperlinks in the hyperlinked documents and document structure to enhance classification performance, we only use the main content as a feature of the hyperlinked document. Therefore, the content extraction or noisy elimination method is one of main issues. We employed a redundant word/phrase noisy elimination method, which is discussed in [20].

A case is defined by attributes as follows:

$CASE = T \cup B$

, where T is a distinct word set of hyperlink text and B is a distinct word set of the main content of the linked document. T and B are respectively defined as $T = \{t_1, t_2, \ldots, t_N\}$ and $B = \{b_1, b_2, \ldots, b_M\}$, where N and M are the number of distinct word and N, M is greater that 0 (N, M $\geq$ 0). t_i and b_j are a word in the hyperlink text and the main text of the hyperlinked document.

A rule structure is defined as follows:

IF

$(TC \subset T)\ AND\ (BC \subset B)\ AND\ (AC \subset T\ or\ AC \subset B)$

THEN

Classify into folder F_i

, where *TC* is a condition set for the hyperlink text, *BC* is a condition set for the hyperlinked document, and *AC* is a condition set for the hyperlink text or the hyperlinked document. Each set is defined as $TC = \{tc_1, tc_2, \ldots, tc_X\}$, $BC = \{bc_1, bc_2, \ldots, bc_Y\}$, and $AC = \{ac_1, ac_2, \ldots, ac_Z\}$, where tc_i is the word in the hyperlink text, bc_j is the word in the hyperlinked document, and ac_k is the word either in the hyperlink text or in the hyperlinked document. The number of words in each condition is greater than 0 (X, Y, Z $\geq$ 0).

In the inference process, the MCRDR-Classifier evaluates each rule node of the knowledge base (KB). If a case is selected from the case list (CL), the system evaluates rules from the root node and the inference result is provided by the last rule satisfied in a pathway. All children of the satisfied parent rule are evaluated, allowing for multiple conclusions. The conclusion of the parent rule is only given if none of the children are satisfied [10, 11]. The MCRDR method does not define any specific rule base tree traversal algorithms, which depends on the implementation strategy. The algorithm on which the MCRDR-Classifier is based is [19] and [11].

6 Knowledge Acquisition of the MCRDR- Classifier

The problem solver performs KA when a case has been classified incorrectly or is missing a classification. The KA editor of the MCRDR-Classifier that is used for KA process is illustrated in Fig. 4. The Knowledge Base (❶) displays the current knowledge base structure which is automatically maintained by the system. The location of the rule is decided according to the rule type. If there is no classification recommendation and the problem solver wants to classify the current case, then the rule is created under the root rule. If the problem solver wants to correct current recommendation by using the refining rule and the stopping rule, the new rule is added at the end of current firing rule to give new classification [19]. The classification structure (❷) used in the MCRDR-Classifier is the traditional folder structure. Any folder can be chosen to specify correct classification and there are two action options. The "select" option is used to create a refining rule and the "deselect" option is to create stopping rule. The case viewer (❺) displays case data handled by the KA editor. Case attributes are displayed in the case attribute viewer (❹). Conditions that are chosen for the new rule are displayed in the condition editor (❸). Whenever new condition words are added, cases that are fired by the current rule are displayed in the cases satisfied rules viewer (❼). These cases are validation cases, and the problem solver does not want to classify some of these cases. However, it is difficult to find conditions that exclude them. To make KA convenient, the system generates different word lists in the case attribute viewer whenever the excluding cases are selected (❻). Therefore, the attributes in the case attribute viewer are not the current case's attributes but the different attributes of current case compared to those of the chosen excluding cases. Unlike the traditional MCRDR systems, the MCRDR-Classifier uses not only cornerstone cases but also classified cases that are fired with new rule to generate a different list in the validation process.

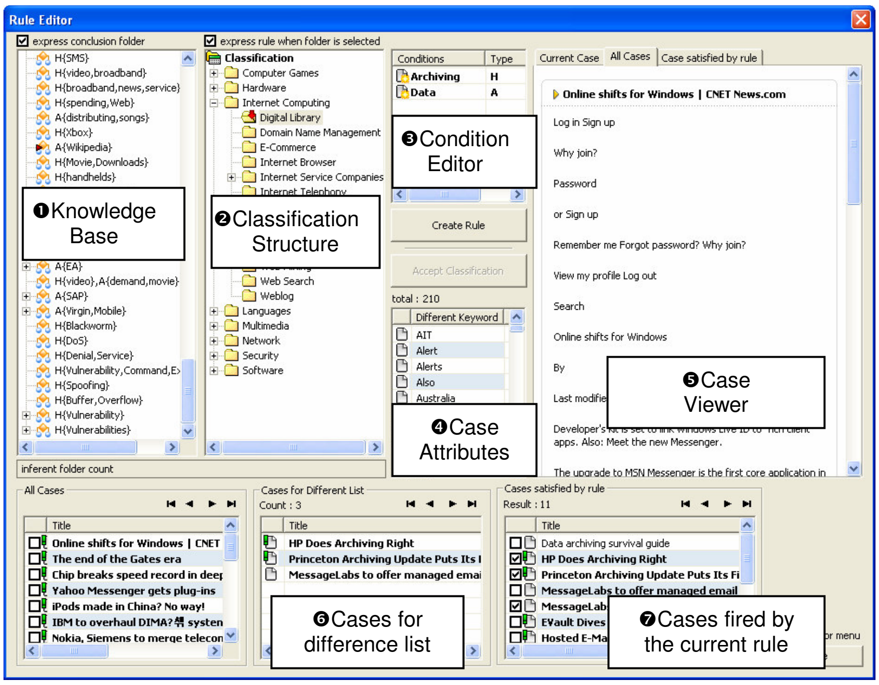


Fig. 4. Knowledge Acquisition Editor of the MCRDR-Classifier

7 Experiment Design

The experiment is designed to analyze KA behaviors in the real-world document classification, an open-ended problem. This experiment was conducted by 20 Master and Hounours course students at the University of Tasmania for four months from August, 2005 ~ November, 2005. The WebMon system continually collected newly updated documents from 9 well known news Websites, which were focused on information technology. Each participant could read the collected documents in real-time and train their own MCRDR-Classifiers. The classification structure (folder structure) of 86 folders was predefined for the experimental purpose. Each participant might use part of all folders for the classification, not the whole classification structure, which totally depended on each participant's intention. In addition, the participants used different number of documents based on his/her document filtering level and created different KB by using his/her own MCRDR-Classifier.

8 Experiments Results

8.1 Overall Classification Results

In total 12,784 articles were collected during the experiment period. In overall 95.6% documents (12,304) were used by the participants. The lowest ratio was 87.6% (BBC) and the highest ratio was 99.8% (ITNews). 13.0% documents were commonly used by all participants, which varied form 5.1% to 22.4%. Compared to the ratio of classified documents, the ratios of commonly used documents are very row. This is caused by the differences in the participant's document filtering level, which will be further illustrated in Section 8.2. Table 1. summarizes monitoring Website list and data set that used in this experiment.

Table 1. Monitoring Websites and Data Set used in the Experiment

Web Site	Monitoring Duration	Number of Docs	Classified Docs	%	Commonly used Docs	%
Australian	116	1,949	1,917	98.4%	437	22.4%
ITnews	116	1,646	1,643	99.8%	212	12.9%
ZDNet	116	1,028	1,024	99.6%	226	22.0%
CNN	116	409	380	92.9%	72	17.6%
BBC	82	651	570	87.6%	54	8.3%
eWEEK	81	1,838	1,795	97.7%	209	11.4%
Sacbee	81	1,228	1,152	93.8%	100	8.1%
CNet	81	3,248	3,065	94.4%	307	9.5%
ITWorld	81	787	758	96.3%	40	5.1%
Total	97[1]	12,784	12,304	95.6%	1,657	13.0%

Note 1. Monitoring duration is a mean average.

8.2 Classification Results by the Participants

Used Documents. Each participant classified documents into multiple folders by using the MCRDR-Classifier. Though the same monitored documents were provided to the all participants, the document usage results are very different. Fig. 5. shows that the number of classified documents differ amongst participants, the smallest number being 1,775 and the largest number 21,045. The mean number is 11,693. The differences caused by two factors. Firstly, ***the document filtering levels*** are different among the participants. That is, the documents that each participant felt sufficiently important to classify were different among participants. Secondly, the ***multiple classifications of each participant*** were also different among the participants because some participants tended to classify document multiple classifications whereas others did it inversely.

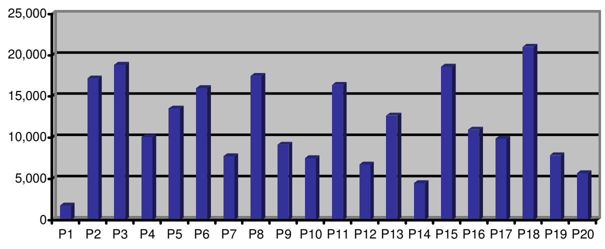
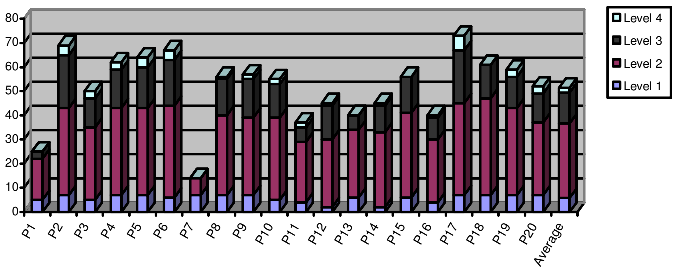
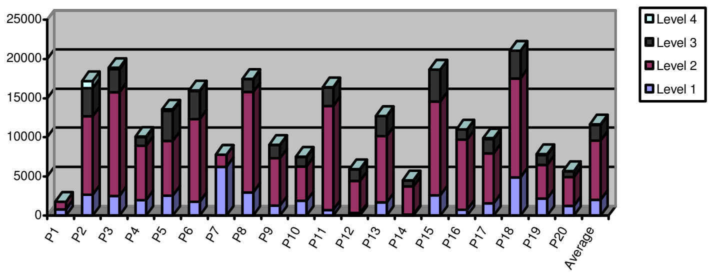


Fig. 5. Number of Classified Documents

Used Folders. A total 86 hierarchical folders were used for this experiment: level one (8), level two (40), level three (31), and level four (8). The numbers of folders that were used for classification were different among participants. They varied from 14 to 73, with the mean number of used folders being 51.35. The most used folders were in level 2, and the least used folders in level 4. There is no evident of the symmetric relationship between the number of used folder and the number of classified documents. For example, though P8, P9, and P10 used a very similar number of folders their correspondent document use is very different (see Fig. 5 (b)).



(a) Folder Usage



(b) Documents in Folders

Fig. 6. Folder Usage by Participants

8.3 Knowledge Acquisition

Rules. On average, the participants created 254 rules for 51 folders with 579 conditions to classify 11,693 documents. The minimum number of rule created was 59 (P13) and the maximum (P18, P19) 597. Documents per rule were 62, rules per folder numbered 5.3, with conditions per rule being 2.3. To examine relationships between document classification and rule creation, and between folder creation and rule creation, correlation values were calculated. The correlation between document classification and rule creation ($CR_{d,r}$) was 0.27 and folder (class) creation and rule creation ($CR_{f,r}$) 0.49.

Conditions. Participants were able to use three different types of condition words, which were seen in title (Type 1), seen in body (Type 2), and seen in both title and body (Type 3). Fig. 8. (a) illustrates each participant's condition usage ratio of three types of rules. Condition selecting depends on each participant's rule construction strategy. Whereas some participants mainly used title condition words (P5, P20), others frequently employed all conditions words (P7, P10, P17).

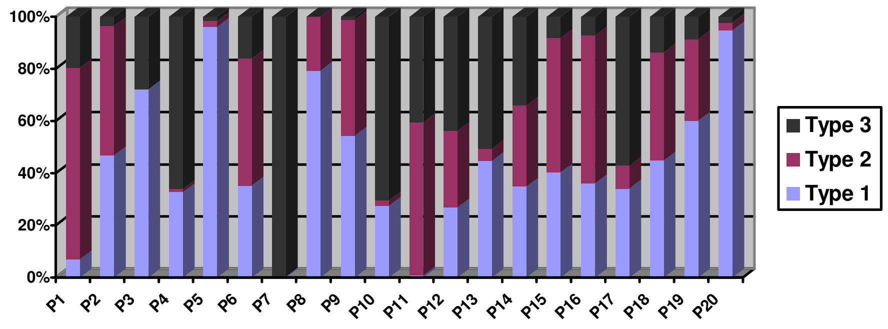


Fig. 8. Participants' Condition Usage Comparison

8 Conclusions

A new problem solving approach is required for open-end problems since the open-ended problems differ from the close-ended problems. Real-world document classification is an open-ended problem because there are no pre-defined classes and cases, and it is possible to classify cases with various coexisting document classifications. We firstly examined the open-ended education experiences to obtain insights into this matter. Active roles of the problem solver, multiple solutions for the same problem and negotiating interactions between the problem solver and the learner were extracted. The MCRDR knowledge acquisition method were employed to implement an open-ended document classification system, called the MCRDR-Classifier. The knowledge base evidently evolves incrementally by employing rule-exception based knowledge representation. The domain experts, or even learners, can easily acquire or construct domain knowledge by using cornerstone cases and difference list. The MCRDR-Classifier supports not only multiple explanations (cases are classified into one class because of different reasons) but also multiple classifications (a case is classified into multiple classes). We conducted experiments to examine knowledge acquisition behaviors. Twenty participants used the MCRDR-Classifier to classify real-world documents. The experiment results show that the participants have different problem solving approaches for the open-ended document classification problem.

References

1. Hong, N.S., *The Relationship Between Well-Structured and Ill-Structured Problem Solving in Multimedia Simulation*, in *The Graduate School, College of Education*. 1998, The Pennsylvania State University.
2. Lynch, C.F., et al. *Defining "Ill-Defined Domains"; A literature survey*. in *Intelligent Tutoring Systems for Ill-Defined Domains*. 2006. Jhongli, Taiwan.
3. Goel, V. *Comparison of Well-Structured & Ill-Structured Task Environments and Problem Spaces*. in *Fourteenth Annual Conference of the Cognitive Science Society*. 1992. Hillsdale, NJ: Erlbaum.
4. Cook, J., *Bridging the Gap Between Empirical Data on Open-Ended Tutorial Interactions and Computational Models.* International Journal of Artificial Intelligence in Education, 2001. **12**: p. 85-99.
5. Andriessen, J. and J. Sandberg, *Where is Education Heading and How About AI?* International Journal of Artificial Intelligence in Education, 1999. **10**: p. 130-150.
6. Mildred, L., G. Shaw, and J.B. Woodward, *Modelling Expert Knowledge*, in *Reading in Knowledge Acquisition and Learning: Automating the Construction and Improvement of Expert Systems*, B.G. Buchanan and D.C. Wilkins, Editors. 1993, Morgan Kaufmann Publishers: San Mateo, CA. p. 77-91.
7. Compton, P. and R. Jansen, *A philosophical basis for knowledge acquisition.* Knowledge Acquisition, 1990. **vol.2, no.3**: p. 241-258.
8. Dillon, R.F. and R.R. Schmeck, *Individual Differences in Cognition*. Vol. 1. 1983, New York, USA: Academic Press, Inc.
9. Blandford, A.E., *Teaching through Collaborative Problem Solving.* Journal of Artificial Intelligence in Education, 1994. **5**(1): p. 51-84.
10. Kang, B., P. Compton, and P. Preston. *Multiple Classification Ripple Down Rules : Evaluation and Possibilities.* in *9th AAAI-Sponsored Banff Knowledge Acquisition for Knowledge-Based Systems Workshop*. 1995. , Banff, Canada, University of Calgary.
11. Compton, P. and D. Richards, *Generalising ripple-down rules.* Knowledge Engineering and Knowledge Management Methods, Models, and Tools. 12th International Conference, EKAW 2000. Proceedings (Lecture Notes in Artificial Intelligence Vol.1937), 2000: p. 380-386.

12. Compton, P., et al., *Knowledge acquisition without analysis.* Knowledge Acquisition for Knowledge-Based Systems. 7th European Workshop, EKAW '93 Proceedings, 1993: p. 277-299.

13. Kang, B.H., P. Compton, and P. Preston, *Validating incremental knowledge acquisition for multiple classifications.* Critical Technology: Proceedings of the Third World Congress on Expert Systems, 1996: p. 856-868.

14. Compton, P. and R. D. *Extending Ripple-Down Rules*. in *12th International Conference on Knowledge Engineering and Knowledge Managements (EKAW'2000)*. 2000. Juan-les-Pins, France.

15. Martinez-Bejar, R., et al., *An easy-maintenance, reusable approach for building knowledge-based systems: application to landscape assessment.* Expert Systems with Applications, 2001. **vol.20, no.2**: p. 153-162.

16. Kang, B.H., W. Gambetta, and P. Compton, *Verification and validation with ripple-down rules.* International Journal of Human-Computer Studies, 1996. **vol.44, no.2**: p. 257-269.

17. Park, S.S., et al. *Automated Information Mediator for HTML and XML based Web Information Delivery Service*. in *The 18th Australian Joint Conference on Artificial Intelligence*. 2005. Sydney, Australia.

18. Park, S.S., S.K. Kim, and B.H. Kang. *Web Information Management System: Personalization and Generalization*. in *the IADIS International Conference WWW/Internet 2003*. 2003.

19. Byeong Ho, K., *Validating Knowledge Acquisition: Multiple Classification Ripple Down Rules*, in *School of Computer Science and Engineering*. 1995, University of New South Wales.

20. Kim, Y.S., S.S. Park, and B.H. Kang. *Noise Elimination from the Web Documents by Using URL Paths and Information Redundancy*. in *IKE'06 - The 2006 International Conference on Information and Knowledge Engineering*. 2006. Monte Carlo Resort, Las Vegas, Nevada, USA: CSREA Press.

Appendix 3. System Setup Manual

SCHOOL OF COMPUTING, UNIVERSITY OF TASMANIA

MCRDR RESEARCH GROUP

iWeb Suite^{TM} User Manual (Version 1.0)

Contents

1 Introduction

1.1 Aims

The iWeb Suite™ is a solution for the Web information management, which integrates Web information change tracking, incremental document classification, and automated Web portal generation technology. This manual provides system management information about the iWeb Suite™ administration in the Windows and Linux environment.

1.2 Users

This manual can be used by

- Domain expert
- System administrator
- Developer

1.3 Contents

This manual aims to provide some basic information about the iWeb Suite™. This manual includes following topics:

- System Architecture
- System Installation and Configuration
- Operation guidelines.

2 System Architecture

2.1 System Overview

The iWeb Suite™ consists of three sub-systems – iWebServer, iWebClient, and iWebPortal. Figure 1 illustrates the overall system architecture of the iWeb Suite™

The **iWebServer ™** collects new information from the registered Web sites and saves it to the database. Detailed explanation about this system will be given Section 4.

The **iWebClient ™** supports incremental document classification for the domain users. Each domain users who receive the monitoring results can use this system independently. Detailed explanation about this system will be given Section 5.

The **iWebPortal ™** supports easy construction of the web portal by using the **iWebClient**. Detailed explanation about this system will be given Section 6.

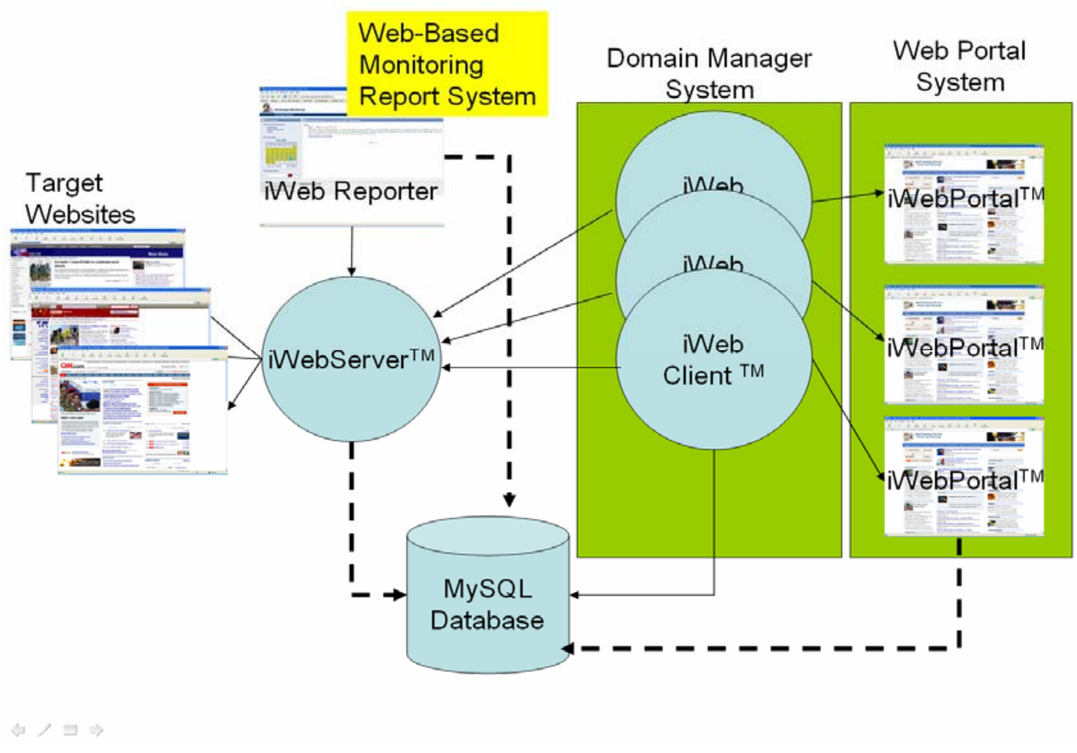


Figure 1. iWeb Suite™ Architecture

2.2 System Environment

OS

The iWebServer™ and the iWebClient™ run on the Microsoft Windows operating system (2000 Server and XP). These two systems can be operated in one system or in the separated system. One iWebServer™ can be used by multiple iWebClient™ systems.

The iWebPortal™ runs on any systems. However, we only tested with Apache Web server in the Linux environment.

MySQL can be installed at any operating system environments.

Database

The iWeb Suite™ uses MySQL (Version 4.1XXX) database for its data storage. You can find more information about the MySQL is in its official Web site (http://www.mysql.com).

Server Side Script Language

The iWebPortal™ is developed by using PHP (Version 5.0.2). You can find more information about the PHP is in its official Web site (http://www.php.net).

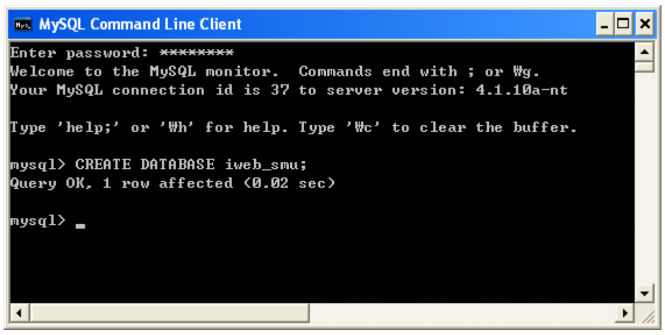
Web Server

The iWebPortal ™ use Apache Web Server (Version 2.0). You can find more information about the Apache Web Server is in its official Web site (http://httpd.apache.org/).

3 System Installation and Configuration

3.1 Database Creation for the iWeb Suite™

You need to create a database that will be used by the iWeb Suite™. You can create a database by using "create database DATABASE_NAME" in the command line. Tables that used for the iWeb Suite™ will be automatically created when you create a "DOMAIN" in the iWebServer (see the 3.4 Domain Registration)

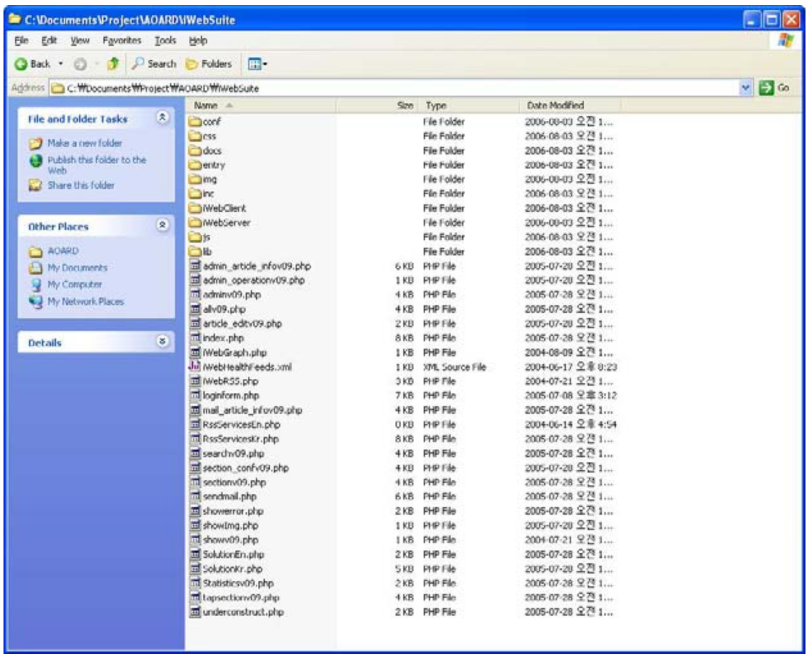


```
Enter password: ********
Welcome to the MySQL monitor.  Commands end with ; or ₩g.
Your MySQL connection id is 37 to server version: 4.1.10a-nt

Type 'help;' or '₩h' for help. Type '₩c' to clear the buffer.

mysql> CREATE DATABASE iweb_smu;
Query OK, 1 row affected (0.02 sec)

mysql> _
```

Figure 2. Database Creation

3.2 iWeb Suite™ Installation

Unzip the WebSuite.zip file in the installation CD and copy the extracted files (see Figure 3)into the "Document Root" directory of Apache.

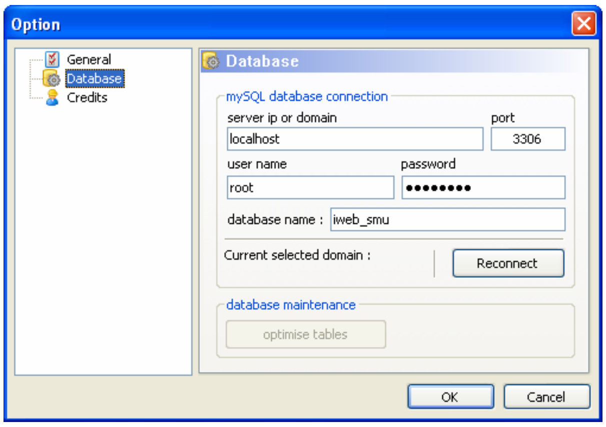


Name	Size	Type	Date Modified
conf		File Folder	2006-08-03 오전 1...
css		File Folder	2006-08-03 오전 1...
docs		File Folder	2006-08-03 오전 1...
entry		File Folder	2006-08-03 오전 1...
img		File Folder	2006-08-03 오전 1...
inc		File Folder	2006-08-03 오전 1...
iWebClient		File Folder	2006-08-03 오전 1...
iWebServer		File Folder	2006-08-03 오전 1...
js		File Folder	2006-08-03 오전 1...
lib		File Folder	2006-08-03 오전 1...
admin_article_infov09.php	6 KB	PHP File	2005-07-28 오전 1...
admin_operationv09.php	1 KB	PHP File	2005-07-28 오전 1...
adminv09.php	4 KB	PHP File	2005-07-28 오전 1...
allv09.php	4 KB	PHP File	2005-07-28 오전 1...
article_editv09.php	2 KB	PHP File	2005-07-28 오전 1...
index.php	8 KB	PHP File	2005-07-28 오전 1...
iWebGraph.php	1 KB	PHP File	2004-08-09 오전 1...
iWebHealthFeeds.xml	1 KB	XML Source File	2004-06-17 오후 8:23
iWebRSS.php	3 KB	PHP File	2004-07-21 오전 1...
loginform.php	7 KB	PHP File	2005-07-08 오후 3:12
mail_article_infov09.php	4 KB	PHP File	2005-07-28 오전 1...
RssServicesEn.php	0 KB	PHP File	2004-06-14 오후 4:54
RssServicesKr.php	8 KB	PHP File	2005-07-28 오전 1...
searchv09.php	4 KB	PHP File	2005-07-28 오전 1...
section_confv09.php	4 KB	PHP File	2005-07-28 오전 1...
sectionv09.php	4 KB	PHP File	2005-07-28 오전 1...
sendmail.php	6 KB	PHP File	2005-07-28 오전 1...
showerror.php	2 KB	PHP File	2005-07-28 오전 1...
showImg.php	1 KB	PHP File	2005-07-28 오전 1...
showv09.php	1 KB	PHP File	2004-07-21 오전 1...
SolutionEn.php	2 KB	PHP File	2005-07-28 오전 1...
SolutionKr.php	5 KB	PHP File	2005-07-28 오전 1...
Statisticsv09.php	2 KB	PHP File	2005-07-28 오전 1...
topsectionv09.php	4 KB	PHP File	2005-07-28 오전 1...
underconstruct.php	2 KB	PHP File	2005-07-28 오전 1...

Figure 3. Extracted Files

3.3 iWebServer™ Configuration

1. Run the iWebServer™ execution file (DOCUMENT_ROOT/iWebServer/MonServ.exe).
2. Close "database connection error" dialog box.
3. Select "Option" menu (Tool>Option) to configure database
4. Select "Database" option from the option list.
5. Fill in the database information – Database Server IP, PORT number (default: 3306), database username, database password, database name.
6. Check the configuration by clicking "Reconnect" button.
7. If there is no problem, click "OK" button and restart the iWebServer™.



server ip or domain	localhost
port	3306
user name	root
password	••••••••
database name :	iweb_smu
Current selected domain :	

Figure 4. Database Configuration

Note: Sometimes when you finish the iWebServer™, the system still run in the background and incur execution problems of the iWebServer™. In that case, you should run the "Window Task Manager" program and end the "MonServ.exe" program and restart the iWebServer™ (see Figure 5)

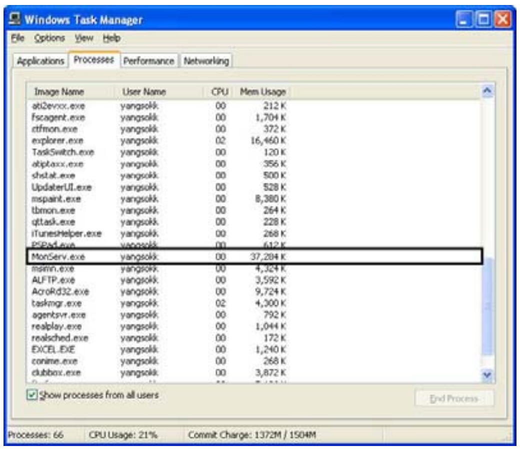


Image Name	User Name	CPU	Mem Usage
ati2evxx.exe	yangsokk	00	212 K
fscagent.exe	yangsokk	00	1,704 K
ctfmon.exe	yangsokk	00	372 K
explorer.exe	yangsokk	02	16,460 K
TaskSwitch.exe	yangsokk	00	120 K
atiptaxx.exe	yangsokk	00	356 K
shstat.exe	yangsokk	00	500 K
UpdaterUI.exe	yangsokk	00	528 K
mspaint.exe	yangsokk	00	8,380 K
tbmon.exe	yangsokk	00	264 K
qttask.exe	yangsokk	00	228 K
iTunesHelper.exe	yangsokk	00	268 K
PSPad.exe	yangsokk	00	612 K
MonServ.exe	yangsokk	00	37,284 K
msmn.exe	yangsokk	00	4,324 K
ALFTP.exe	yangsokk	00	3,592 K
AcroRd32.exe	yangsokk	00	9,724 K
taskmgr.exe	yangsokk	02	4,300 K
agentsvr.exe	yangsokk	00	792 K
realplay.exe	yangsokk	00	1,044 K
realsched.exe	yangsokk	00	172 K
EXCEL.EXE	yangsokk	00	1,240 K
conime.exe	yangsokk	00	268 K
clubbox.exe	yangsokk	00	3,872 K

[✓] Show processes from all users

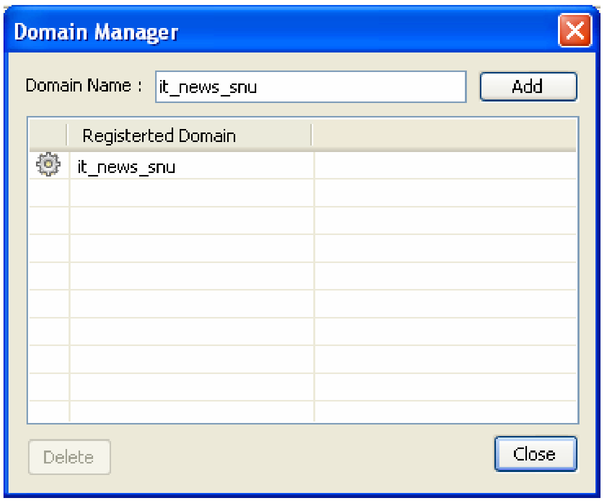
Processes: 66 CPU Usage: 21% Commit Charge: 1372M / 1504M

Figure 5. End the "MonServ.exe" in the Window Task Manager

3.4 Domain Registration

The “Domain” is the top level concept of information management in the iWeb Suite™. User can choose any domain name. For example, if you want to collect and manage information technology information from the Web, you can create domain name like “it_new”. The domain will be used as part of the table name (e.g., tbclassify_DOMAIN_NAME_article). The domain name should be lower case and put the under when you want to make space in the domain name. When you register domain name, the iWebServer™ automatically create the tables that is used by other system of iWeb Suite™ (see Figure 7).

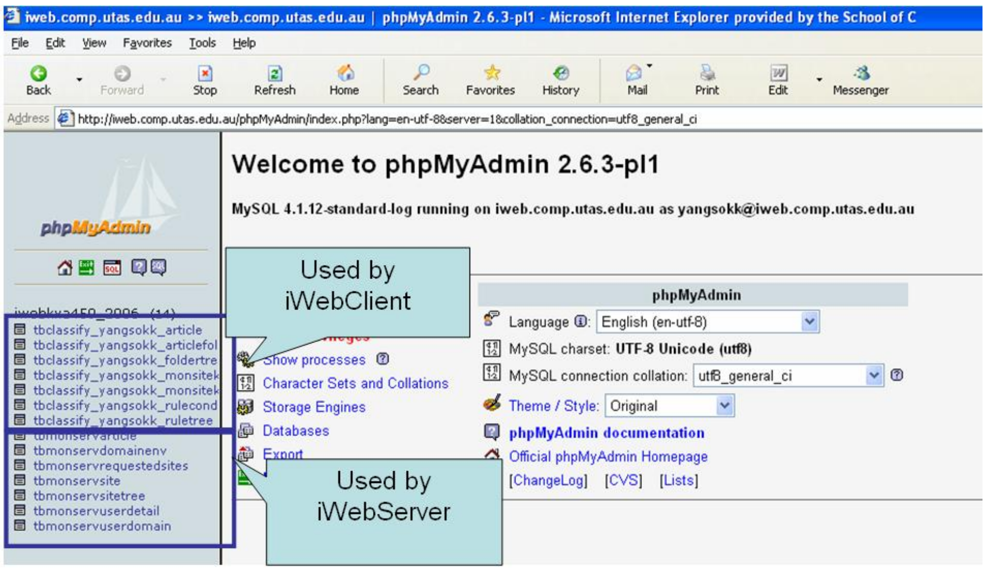
1. Select Tool>Domain Manager from the iWebServer™ menu bar
2. Fill the domain name and click the “Add” button.


Domain Manager

Domain Name : it_news_snu

Registerted Domain
it_news_snu

Figure 6. Domain Creation


iweb.comp.utas.edu.au >> iweb.comp.utas.edu.au | phpMyAdmin 2.6.3-pl1 - Microsoft Internet Explorer provided by the School of C

Address http://iweb.comp.utas.edu.au/phpMyAdmin/index.php?lang=en-utf-8&server=1&collation_connection=utf8_general_ci

Welcome to phpMyAdmin 2.6.3-pl1

MySQL 4.1.12-standard-log running on iweb.comp.utas.edu.au as yangsokk@iweb.comp.utas.edu.au

tbclassify_yangsokk_article
tbclassify_yangsokk_articlefol
tbclassify_yangsokk_foldertre
tbclassify_yangsokk_monsitek
tbclassify_yangsokk_monsitek
tbclassify_yangsokk_rulecond
tbclassify_yangsokk_ruletree
tbmonservarticle
tbmonservdomainenv
tbmonservrequestedsites
tbmonservsite
tbmonservsitetree
tbmonservuserdetail
tbmonservuserdomain

phpMyAdmin

Language: English (en-utf-8)
MySQL charset: UTF-8 Unicode (utf8)
MySQL connection collation: utf8_general_ci
Theme / Style: Original

File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Print Edit Messenger
phpMyAdmin
Used by iWebClient
Used by iWebServer
Show processes
Character Sets and Collations
Storage Engines
Databases
Export
phpMyAdmin documentation
Official phpMyAdmin Homepage
[ChangeLog] [CVS] [Lists]

Figure 7. Database Structure after “yangsokk” domain is created

3.5 iWebClient™ Configuration

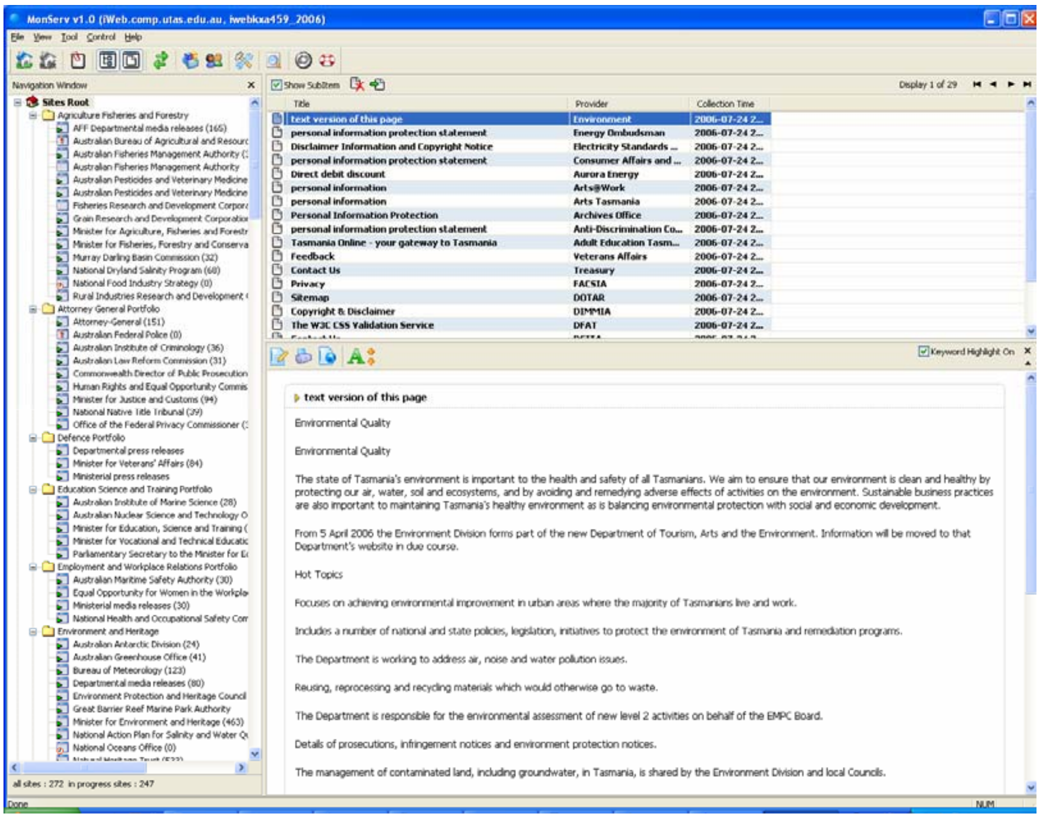
The configuration of the iWebClient™ is very similar to that of the iWebServer™.

1. Run the iWebClient™ execution file (DOCUMENT_ROOT/iWebClient/MonClassifier.exe).
2. Close "database connection error" dialog box.
3. Select "Option" menu (Tool>Option) to configure database
4. Select "Database" option from the option list.
5. Fill in the database information – Database Server IP, PORT number (default: 3306), database username, database password, database name.
6. Check the configuration by clicking "Reconnect" button.
7. If there is no problem, click "OK" button and restart the iWebClient™.

4 iWebServer TM

4.1 System User Interface

Figure 6 illustrates the main user interface of the iWebServer system. The left pane displays the registered Web sites, which are grouped by specific topics (e.g., Agriculture Fisheries and Forestry). The right upper pane shows newly collected documents' title, and the right bottom pane presents text of the selected documents. This system differs from the RSS service because it does not need any specific XML formats and it is based on automated client pull technology. Newly collected documents can be accessed by using the iWebClient™.


MonServ v1.0 (iWeb.comp.utas.edu.au, iwebkxa459_2006)
File View Tool Control Help
Navigation Window
Show SubItem
Display 1 of 29
Keyword Highlight On
Done
NUM

Sites Root
- Agriculture Fisheries and Forestry
  - AFF Departmental media releases (165)
  - Australian Bureau of Agricultural and Resourc
  - Australian Fisheries Management Authority (:
  - Australian Fisheries Management Authority
  - Australian Pesticides and Veterinary Medicine
  - Australian Pesticides and Veterinary Medicine
  - Fisheries Research and Development Corpor
  - Grain Research and Development Corporatior
  - Minister for Agriculture, Fisheries and Forestr
  - Minister for Fisheries, Forestry and Conserva
  - Murray Darling Basin Commission (32)
  - National Dryland Salinity Program (68)
  - National Food Industry Strategy (0)
  - Rural Industries Research and Development (
- Attorney General Portfolio
  - Attorney-General (151)
  - Australian Federal Police (0)
  - Australian Institute of Criminology (36)
  - Australian Law Reform Commission (31)
  - Commonwealth Director of Public Prosecution
  - Human Rights and Equal Opportunity Commis
  - Minister for Justice and Customs (94)
  - National Native Title Tribunal (39)
  - Office of the Federal Privacy Commissioner (:
- Defence Portfolio
  - Departmental press releases
  - Minister for Veterans' Affairs (84)
  - Ministerial press releases
- Education Science and Training Portfolio
  - Australian Institute of Marine Science (28)
  - Australian Nuclear Science and Technology O
  - Minister for Education, Science and Training (
  - Minister for Vocational and Technical Educatic
  - Parliamentary Secretary to the Minister for E
- Employment and Workplace Relations Portfolio
  - Australian Maritime Safety Authority (30)
  - Equal Opportunity for Women in the Workpla
  - Ministerial media releases (30)
  - National Health and Occupational Safety Com
- Environment and Heritage
  - Australian Antarctic Division (24)
  - Australian Greenhouse Office (41)
  - Bureau of Meteorology (123)
  - Departmental media releases (80)
  - Environment Protection and Heritage Council
  - Great Barrier Reef Marine Park Authority
  - Minister for Environment and Heritage (463)
  - National Action Plan for Salinity and Water Q
  - National Oceans Office (0)

all sites : 272 in progress sites : 247

Title	Provider	Collection Time
text version of this page	Environment	2006-07-24 2...
personal information protection statement	Energy Ombudsman	2006-07-24 2...
Disclaimer Information and Copyright Notice	Electricity Standards ...	2006-07-24 2...
personal information protection statement	Consumer Affairs and ...	2006-07-24 2...
Direct debit discount	Aurora Energy	2006-07-24 2...
personal information	Arts@Work	2006-07-24 2...
personal information	Arts Tasmania	2006-07-24 2...
Personal Information Protection	Archives Office	2006-07-24 2...
personal information protection statement	Anti-Discrimination Co...	2006-07-24 2...
Tasmania Online - your gateway to Tasmania	Adult Education Tasm...	2006-07-24 2...
Feedback	Veterans Affairs	2006-07-24 2...
Contact Us	Treasury	2006-07-24 2...
Privacy	FACSIA	2006-07-24 2...
Sitemap	DOTAR	2006-07-24 2...
Copyright & Disclaimer	DIMMIA	2006-07-24 2...
The W3C CSS Validation Service	DFAT	2006-07-24 2...

text version of this page

Environmental Quality

Environmental Quality

The state of Tasmania's environment is important to the health and safety of all Tasmanians. We aim to ensure that our environment is clean and healthy by protecting our air, water, soil and ecosystems, and by avoiding and remedying adverse effects of activities on the environment. Sustainable business practices are also important to maintaining Tasmania's healthy environment as is balancing environmental protection with social and economic development.

From 5 April 2006 the Environment Division forms part of the new Department of Tourism, Arts and the Environment. Information will be moved to that Department's website in due course.

Hot Topics

Focuses on achieving environmental improvement in urban areas where the majority of Tasmanians live and work.

Includes a number of national and state policies, legislation, initiatives to protect the environment of Tasmania and remediation programs.

The Department is working to address air, noise and water pollution issues.

Reusing, reprocessing and recycling materials which would otherwise go to waste.

The Department is responsible for the environmental assessment of new level 2 activities on behalf of the EMPC Board.

Details of prosecutions, infringement notices and environment protection notices.

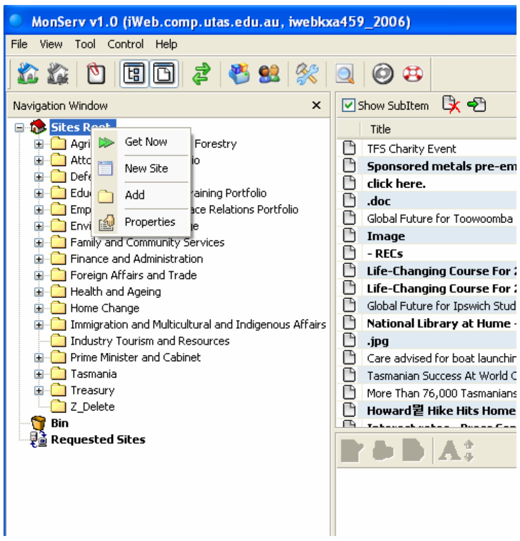
The management of contaminated land, including groundwater, in Tasmania, is shared by the Environment Division and local Councils.

Figure 8. User Interface of iWeb Server

4.2 Web Monitoring Procedure

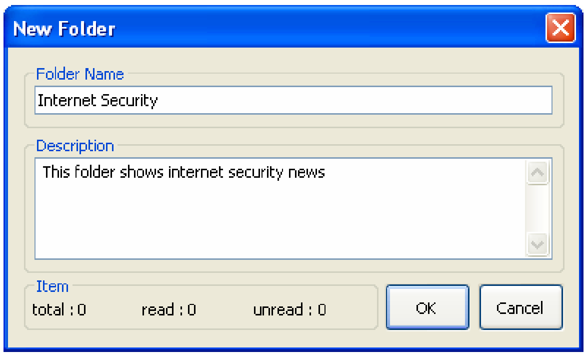
4.2.1 Folder Creation

The iWebServer™ supports grouping of monitoring Web sites. To this end, user can create any folder and add Web site under this folder. To create the folder, select root node of monitoring Web site tree or any folder in this tree and click the short-cut menu for folder creation. When the following "new folder" dialog will appear, the user fill in the name of folder. User can move Web site folder to folder as he/she wishes.


MonServ v1.0 (iWeb.comp.utas.edu.au, iwebkxa459_2006)

Sites Root
- Agri… Forestry
- Attc…
- Defe…
- Edu… …aining Portfolio
- Emp… …ce Relations Portfolio
- Envi…
- Family and Community Services
- Finance and Administration
- Foreign Affairs and Trade
- Health and Ageing
- Home Change
- Immigration and Multicultural and Indigenous Affairs
- Industry Tourism and Resources
- Prime Minister and Cabinet
- Tasmania
- Treasury
- Z_Delete
- Bin
- Requested Sites

Title
TFS Charity Event
Sponsored metals pre-em
click here.
.doc
Global Future for Toowoomba
Image
- RECs
Life-Changing Course For
Life-Changing Course For
Global Future for Ipswich Stud
National Library at Hume
.jpg
Care advised for boat launchir
Tasmanian Success At World C
More Than 76,000 Tasmanians
Howard Hike Hits Home


New Folder

Folder Name: Internet Security

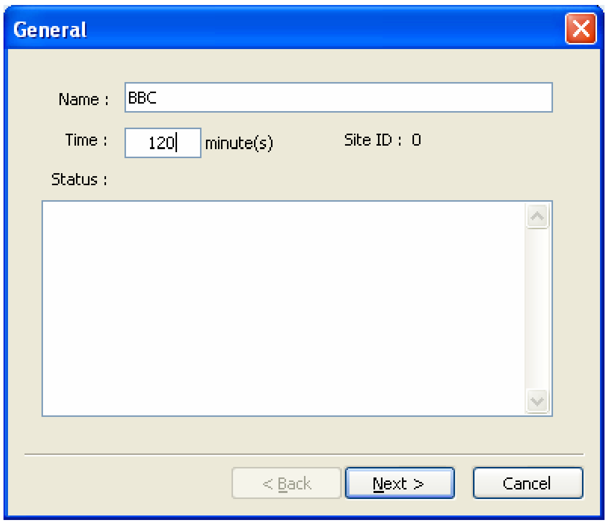
Description: This folder shows internet security news

Item: total : 0 read : 0 unread : 0

Figure 9. Folder Creation

4.2.2 Monitoring Site Registration

1. Click the new monitoring site add icon ( ).
2. Fill in the site name and monitoring interval time. Click “Next” button

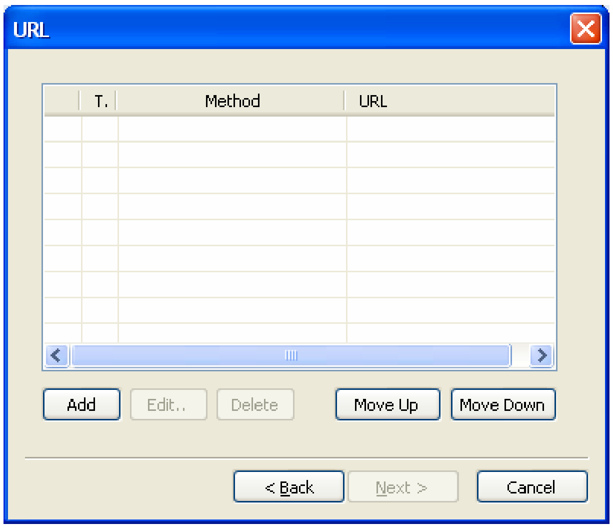

General

Name : BBC

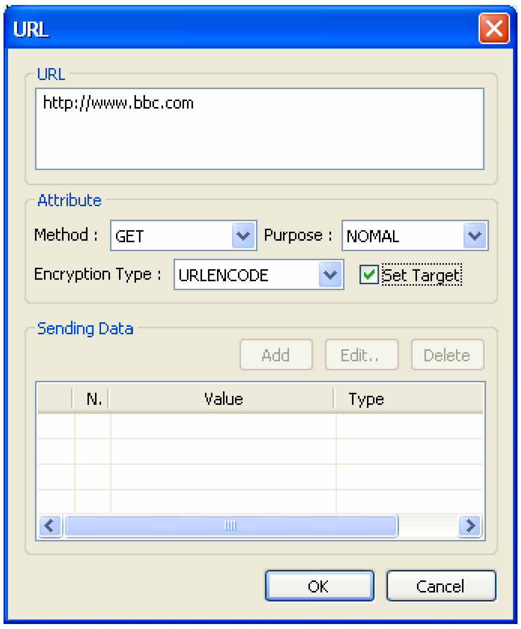
Time : 120 minute(s) Site ID : 0

Status :

3. Click "Add" button

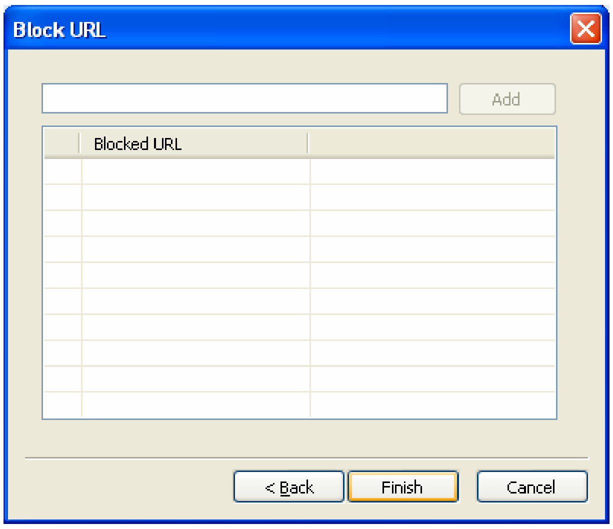


4. File in URL and check "Set Target" option. Click "OK" button.


URL: http://www.bbc.com

Attribute
Method : GET
Purpose : NOMAL
Encryption Type : URLENCODE
[✓ Set Target]

5. Add known block URLs and click "Finish" button.



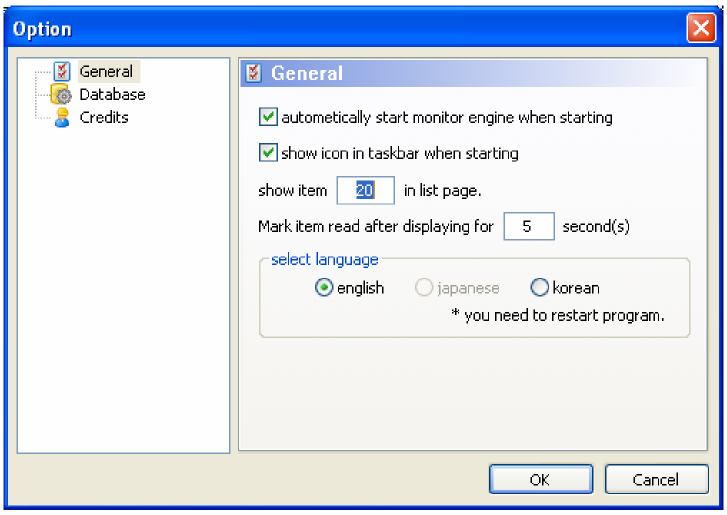
4.2.3 Monitoring Start and Stop

The user can start or stop monitoring individual Web site by individual Web Site or grouped Web sites.

1. Select Web site of Folders
2. Click short-cut menu (see Figure 7) "Get Now" or "Stop"

4.2.4 Automated Monitoring

You can configure to start all Web site monitoring whenever the iWeb Server starts by checking the "automatically start monitor engine when the system starts".



Option

General

[✓] autometically start monitor engine when starting

[✓] show icon in taskbar when starting

show item 20 in list page.

Mark item read after displaying for 5 second(s)

select language

[✓ english] japanese korean

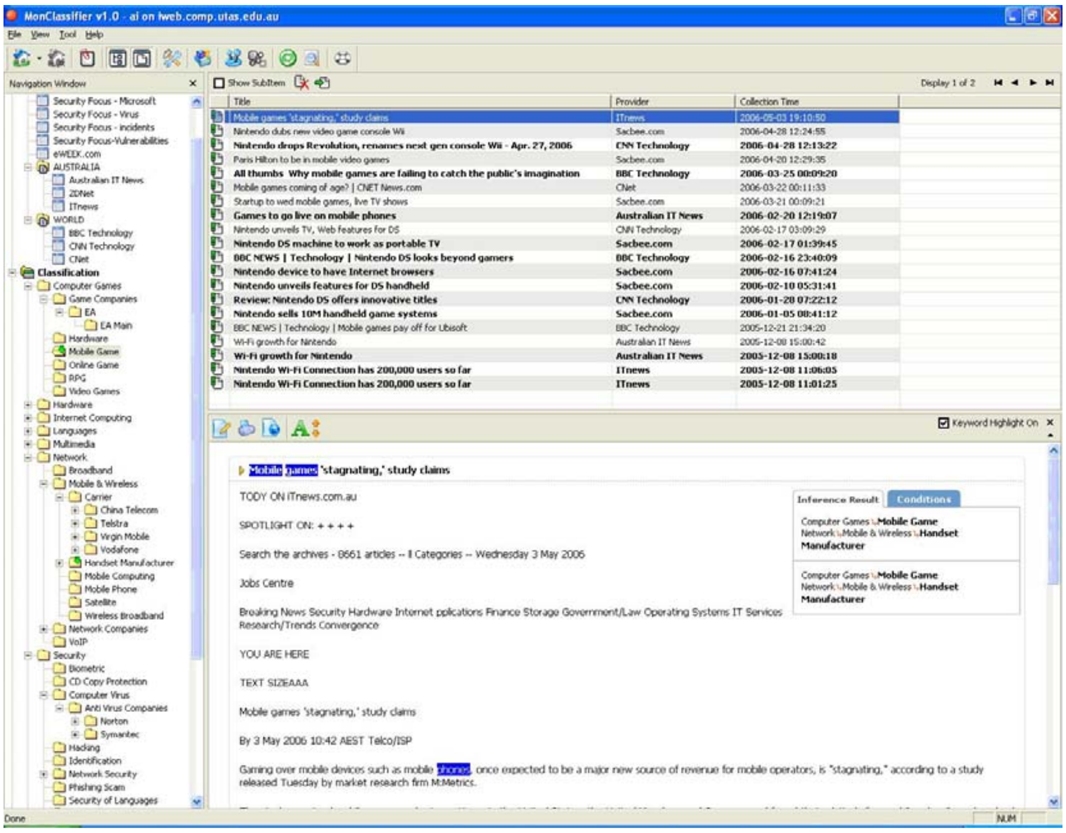
* you need to restart program.

Figure 10. Automatic Start Option

5 iWebClient TM

5.1 System User Interface

Figure 11 illustrates the main user interface of the iWebClient™, which consists of three parts. The left pane shows the tracking Website list (❶) that are selected by the problem solver and the classification structure (❷). The problem solver can choose his/her favorite tracking Websites that are maintained by iWebServer™. When the problem solver selects one Website in the list, newly classified documents that are ordered by collected time are displayed in the right upper pane (❸). When one document is selected, the content of the document the inference results are displayed in the classification structure (❷) by displaying downward arrow (s) and the right top corner of the content pane (❹). The problem solver can see the hierarchical structure of the classification by both the classification folder tree (CFT) and the folder path description in❹. The conditions of fired rule sets are also displayed by selecting the condition tap and also highlighted in the content. More detailed inference procedures are explained in the following section. One important thing is that the iWebClient™ supports multiple classifications. For example, there are two destine folders (Mobile game and Handset Manufacturer) for the document “Mobile games 'stagnating,' study claims”.



Title	Provider	Collection Time
Mobile games 'stagnating,' study claims	ITnews	2006-05-03 19:10:50
Nintendo dubs new video game console Wii	Sacbee.com	2006-04-28 12:24:55
Nintendo drops Revolution, renames next gen console Wii - Apr. 27, 2006	**CNN Technology**	**2006-04-28 12:13:22**
Paris Hilton to be in mobile video games	Sacbee.com	2006-04-20 12:29:35
All thumbs Why mobile games are failing to catch the public's imagination	**BBC Technology**	**2006-03-25 00:09:20**
Mobile games coming of age? | CNET News.com	CNet	2006-03-22 00:11:33
Startup to wed mobile games, live TV shows	Sacbee.com	2006-03-21 00:09:21
Games to go live on mobile phones	**Australian IT News**	**2006-02-20 12:19:07**
Nintendo unveils TV, Web features for DS	CNN Technology	2006-02-17 03:09:29
Nintendo DS machine to work as portable TV	**Sacbee.com**	**2006-02-17 01:39:45**
BBC NEWS | Technology | Nintendo DS looks beyond gamers	**BBC Technology**	**2006-02-16 23:40:09**
Nintendo device to have Internet browsers	**Sacbee.com**	**2006-02-16 07:41:24**
Nintendo unveils features for DS handheld	**Sacbee.com**	**2006-02-10 05:31:41**
Review: Nintendo DS offers innovative titles	**CNN Technology**	**2006-01-20 07:22:12**
Nintendo sells 10M handheld game systems	**Sacbee.com**	**2006-01-05 00:41:12**
BBC NEWS | Technology | Mobile games pay off for Ubisoft	BBC Technology	2005-12-21 21:34:20
Wi-Fi growth for Nintendo	Australian IT News	2005-12-08 15:00:42
Wi-Fi growth for Nintendo	**Australian IT News**	**2005-12-08 15:00:18**
Nintendo Wi-Fi Connection has 200,000 users so far	**ITnews**	**2005-12-08 11:06:05**
Nintendo Wi-Fi Connection has 200,000 users so far	**ITnews**	**2005-12-08 11:01:25**

Mobile games 'stagnating,' study claims

TODY ON iTnews.com.au

SPOTLIGHT ON: + + + +

Search the archives - 8661 articles -- ll Categories -- Wednesday 3 May 2006

Jobs Centre

Breaking News Security Hardware Internet pplications Finance Storage Government/Law Operating Systems IT Services Research/Trends Convergence

YOU ARE HERE

TEXT SIZEAAA

Mobile games 'stagnating,' study claims

By 3 May 2006 10:42 AEST Telco/ISP

Gaming over mobile devices such as mobile phones, once expected to be a major new source of revenue for mobile operators, is "stagnating," according to a study released Tuesday by market research firm M:Metrics.

Inference Result | Conditions

Computer Games \ **Mobile Game**
Network \ Mobile & Wireless \ **Handset Manufacturer**

Computer Games \ **Mobile Game**
Network \ Mobile & Wireless \ **Handset Manufacturer**

Figure 11. Main User Interface of the MCRDR-Classifier

5.2 Inference Process of the iWeb Client TM

Basically, the inference process implementation of the iWebClient™ follows the MCRDR method. In the iWebClient™, the case is each hyperlink text in the monitoring Webpage and hyperlinked document and its attributes are distinct words in each text. The iWebServer™ continually revisits target Websites and finds new hyperlink as newly updated information. The text that is wrapped by <a> tag is used as "Title". Though it may or may not match with the title of the hyperlinked document, it is usually used to represent the hyperlinked document. The hyperlinked document contains not only main content but also noisy or redundant content such as advertisement, navigation or decorative content. Though we can use other features such as Hyperlink and document structure and these features may enhance classification performance, we only use the main content as feature of the hyperlinked document, not use these features in our system. Therefore, one of main issue is the content extraction method and we employed a redundant word/phrase elimination method, which is discussed in the other paper separately.

A case is defined by attributes as follows:

$$CASE = T \cup B$$

, where T is a distinct word set of hyperlink text and B is a distinct word set of the main content of linked document. T and B are respectively defined as follows:

$$T = \{t_1, t_2, \ldots, t_N\}$$

$$B = \{b_1, b_2, \ldots, b_M\}$$

, where N and M is the number of distinct word and N, M is greater that 0 (N, M $\geq$ 0). t_i and b_j is a word in the hyperlink text and the main text of the hyperlinked document.

A rule structure is defined as follows:

```
IF
        $TC \subset T$
        AND $BC \subset B$
        AND ($AC \subset T$ or $AC \subset B$)
THEN
        Classify into folder $F^i$
```

, where TC is a condition set for the hyperlink text, BC is a condition set for the hyperlinked document, and AC is a condition set for the hyperlink text or the hyperlinked document. Each set is defined as follows:

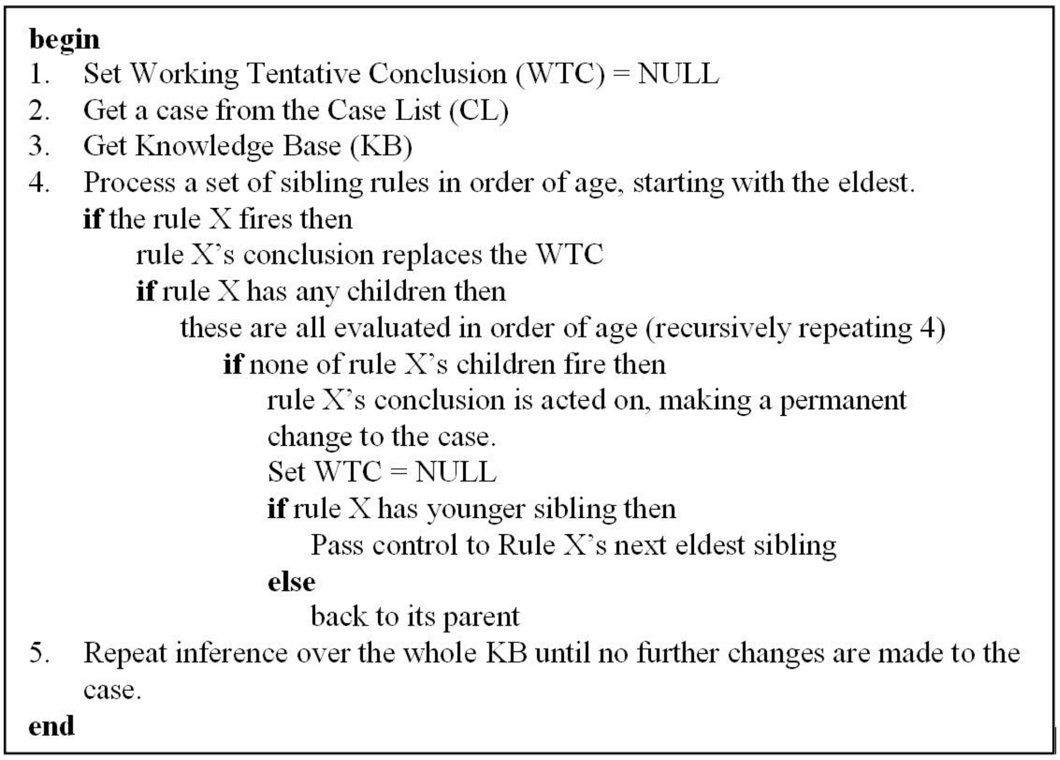
$$TC = \{tc_1, tc_2, \ldots, tc_X\}$$

$$BC = \{bc_1, bc_2, \ldots, bc_Y\}$$

$$AC = \{ac_1, ac_2, \ldots, ac_Z\}$$

, where tc_i is the word in the hyperlink text, bc_j is the word in the hyperlinked document, and ac_k is the word either in the hyperlink text or in the hyperlinked document. The number of word of each condition is greater than 0 ($X, Y, Z \geq 0$).

In the inference process, the iWebClient™ evaluates each rule node of knowledge base (KB). If a case is selected from the case list (CL), the system evaluates rules from the root node and inference result is provided by the last rule satisfied in a pathway. All children of satisfied parent rule are evaluated, allowing for multiple conclusions. The conclusion of the parent rule is only given if none of the children are satisfied. However, the MCRDR method does not define specific rule base tree traversal algorithm, it depends on the implementation strategy. The inference algorithm of the iWebClient™ is explained in Figure 12.

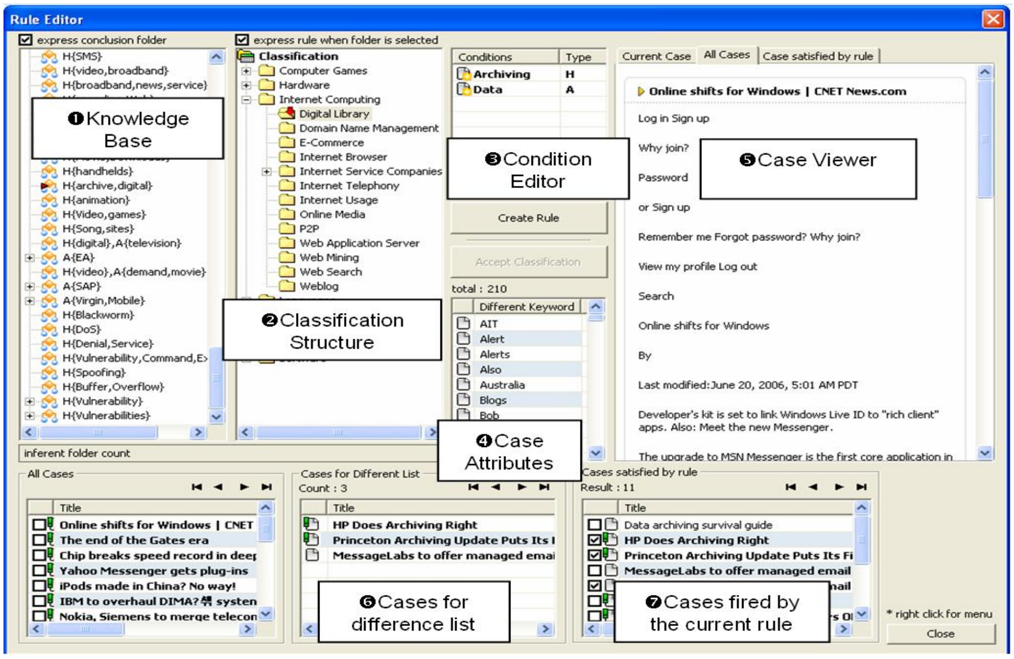


```
begin
1.  Set Working Tentative Conclusion (WTC) = NULL
2.  Get a case from the Case List (CL)
3.  Get Knowledge Base (KB)
4.  Process a set of sibling rules in order of age, starting with the eldest.
    if the rule X fires then
        rule X's conclusion replaces the WTC
        if rule X has any children then
            these are all evaluated in order of age (recursively repeating 4)
                if none of rule X's children fire then
                    rule X's conclusion is acted on, making a permanent
                    change to the case.
                    Set WTC = NULL
                    if rule X has younger sibling then
                        Pass control to Rule X's next eldest sibling
                    else
                        back to its parent
5.  Repeat inference over the whole KB until no further changes are made to the
    case.
end
```

Figure 12. Inference Algorithm in the iWeb Client ™

5.3 Knowledge Acquisition of the iWebClient ™

The problem solver performs KA when **a case has been classified incorrectly or is missing a classification**. The KA editor of the iWebClient™ that used for KA process is illustrated in Figure 13. The Knowledge Base (❶) displays current knowledge base structure, which is maintained by the system. The locations of rules decided according to the rule type. Firstly, when a Type 1 rule (new independent rule) is created, the rule is added under the root rule. Secondly, when a Type 2 rule (refining rule) is created, the rule is added at the end of current firing rule to give new classification. Lastly, when a Type 3 rule (stopping) is created, the rule is added at the end of path to prevent the classification. The classification structure (❷) is traditional folder structure. Any folder can be chosen to specify correct classification and there are two options for choosing. The "select" option is used for Rule Type 1 (new independent rule) and 2(refining rule) and the "deselect" option is for the Rule Type 3 (stopping rule). The case viewer (❺) displays case data that handled by the KA editor. Case attributes are displayed in the case attribute viewer (❹). Conditions that are chosen for the new rule are displayed in the condition editor (❸). Whenever new condition words are added, cases that are fired by the current rule are displayed in the cases satisfied rules viewer (❼). These cases are validation cases. The problem solver does not want to classify some of these cases. However, it is difficult to find conditions that exclude them. To support easy KA, the system generates difference words lists in the case attribute viewer whenever the excluding cases are selected (❻). Therefore, the attributes in the case attribute viewer are not the current case's attributes but the different attributes of current case compared to those of the chosen excluding cases. Unlike the traditional MCRDR systems, the iWebClient™ use not only cornerstone cases but also classified cases that are fired new rule to generate a difference list in the validation process.



☑ express conclusion folder

H{SMS}
H{video,broadband}
H{broadband,news,service}
H{handhelds}
H{archive,digital}
H{animation}
H{Video,games}
H{Song,sites}
H{digital},A{television}
A{EA}
H{video},A{demand,movie}
A{SAP}
A{Virgin,Mobile}
H{Blackworm}
H{DoS}
H{Denial,Service}
H{Vulnerability,Command,E
H{Spoofing}
H{Buffer,Overflow}
H{Vulnerability}
H{Vulnerabilities}

inferent folder count

☑ express rule when folder is selected

Classification
- Computer Games
- Hardware
- Internet Computing
  - Digital Library
  - Domain Name Management
  - E-Commerce
  - Internet Browser
  - Internet Service Companies
  - Internet Telephony
  - Internet Usage
  - Online Media
  - P2P
  - Web Application Server
  - Web Mining
  - Web Search
  - Weblog

Conditions	Type
Archiving	H
Data	A

total : 210

Different Keyword
AIT
Alert
Alerts
Also
Australia
Blogs
Bob

Current Case | All Cases | Case satisfied by rule

Online shifts for Windows | CNET News.com

Log in Sign up

Why join?

Password

or Sign up

Remember me Forgot password? Why join?

View my profile Log out

Search

Online shifts for Windows

By

Last modified:June 20, 2006, 5:01 AM PDT

Developer's kit is set to link Windows Live ID to "rich client" apps. Also: Meet the new Messenger.

The upgrade to MSN Messenger is the first core application in

All Cases

Title
☐ Online shifts for Windows | CNET
☐ The end of the Gates era
☐ Chip breaks speed record in deep
☐ Yahoo Messenger gets plug-ins
☐ iPods made in China? No way!
☐ IBM to overhaul DIMA? system
☐ Nokia, Siemens to merge telecon

Cases for Different List
Count : 3

Title
HP Does Archiving Right
Princeton Archiving Update Puts Its I
MessageLabs to offer managed emai

Cases satisfied by rule
Result : 11

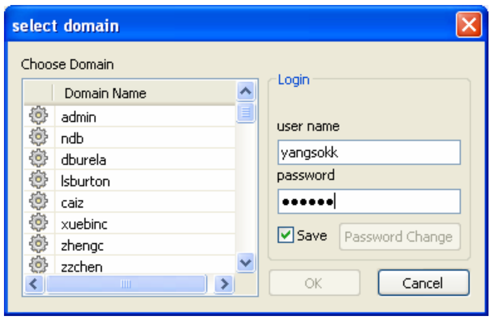
Title
☐ Data archiving survival guide
☑ HP Does Archiving Right
☑ Princeton Archiving Update Puts Its Fi
☐ MessageLabs to offer managed email
☑ [illegible]
☐ [illegible]
☐ [illegible]

Figure 13. Knowledge Acquisition Editor of the iWebClient ™

5.4 Classification Procedure

5.4.1 Login

When you use the iWeb Client, you need to type in username (account name) and password (student ID) and select your domain (e.g., yangsokk). Then click the "OK" button in the "select domain" dialog box and "database configuration" box. When you login is successful, you can see the main user interface (Figure 11).



select domain

Choose Domain

Domain Name
admin
ndb
dburela
lsburton
caiz
xuebinc
zhengc
zzchen

Login

user name: yangsokk

password: ••••••

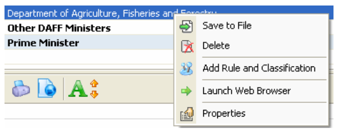
[✓] Save

Figure 14. Login to the iWebClient ™

5.4.2 Knowledge Acquisition

1. When the participant sees the **unclassified or misclassified document**, select it from the document list.

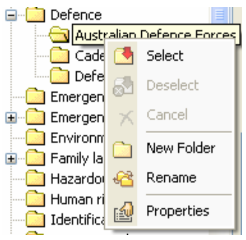
2. Click the KA icon from menu bar or select "Add Rule and Classification" from short cut menu.



Note. In this project, you don't need to create folder in the classification structure. Instead, you only use the pre-defined classification structure.

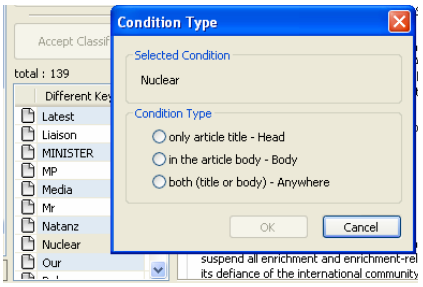
You can see the Knowledge Acquisition interface (Figure 11).

3. Select an appropriate folder from the classification structure by selecting folder and choosing menu from the short-cut menu.

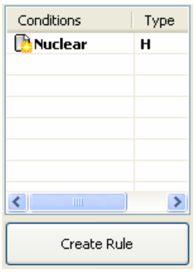


Note. If you want to refine or stop the current rule, select "Deselect" munu from the short-cut menu.

4. Select condition word from the case attribute list. If you select one word from this list, the "Condition Type" dialog box appears. You should choose one of three types (Head, Body, Anywhere).


Accept Classif
total : 139
Different Key
Latest
Liaison
MINISTER
MP
Media
Mr
Natanz
Nuclear
Our
Condition Type
Selected Condition
Nuclear
Condition Type
only article title - Head
in the article body - Body
both (title or body) - Anywhere
OK
Cancel
suspend all enrichment and enrichment-rel
its defiance of the international community

5. If you choose condition type and click "OK" button, this condition appear in the condition editor box. You can add any number of conditions by repeating



Note. If you select conditions, the system retrieves all cases that are fired current condition in the "Cases satisfied by rules" panel. If you don't want to classify some cases in this specified folder, you can choose those cases from this list. The selected cases will be shown in the "Cases for difference list" panel and the difference list is generated in the "Case attribute" panel. If you select condition words from this list, the selected cases will be excluded.

6. If you complete condition editing, then click "Create Rule" button.
7. If you want to classify this case into other folder, repeat Step 3 ~ Step 6. Otherwise, finish knowledge acquisition process by clicking "Close" button.

6 iWebPortal ™

You need to change `config.php` (Document Root/conf/config.php) and `func_db.php` (Document Root/inc/v09/func_db.php) files.

6.1 Domain Name Configuration

You need to change the domain name that you want to use in the portal.

```
//DOMAIN NAME change
$GLOBAL_VARIABLE[Domain] = "it_news";
```

6.2 Database Configuration

You need to change database connection information as that used in the iWebSuite™.

```
function db_connecter()
{
     //change user name and password
     $db_connecter = mysqli_connect("localhost", "user",
"password");
     //change database name
     mysqli_select_db($db_connecter, "iWebEnterpriseEx");
     return $db_connecter;
}
```

6.3 Default Section Configuration

You need to create default section configuration data by using the "Document Root/docs/SectionConfig.sql".

7 Resources and Contact

7.1 PHP Program Language & Libraries

- Official Web Site: http://www.php.net/
- Programming Resources
  - PEAR - PHP Extension and Application Repository
    http://pear.php.net/
  - ZEND – A PHP Company
    http://www.zend.com/
  - PHP Class Repository
    http://www.phpclasses.org/
- Articles & Tutorials
  - DevShed
    http://www.devshed.com/c/b/PHP/
  - PHP Developer
    http://www.phpdeveloper.org/
  - O'Leilly PHP Resource
    http://www.onlamp.com/pub/q/all_php_articles
- Tree Library
  - PHP Layers Menu
    http://phplayersmenu.sourceforge.net/

7.2 MySQL

- Official Site: http://www.mysql.com/

7.3 Contact

- Technical Support: Yang Sok Kim (yangsokk@utas.edu.au)

8 Appendix: Database Tables

(1) Tables used by MonServer

tbmonservarticle Article Data Table for MonServ

Column	Type	Description	Value and Meaning
ArticleID	BIGINT(20)	Article ID Number	Unique ID Number
MonSiteID	BIGINT(20)	Monitoring Site ID Number	Unique ID Number
bRead	BIGINT(20)	Article Read,Unread for Monsitoring Server	0:Unread, 1:Read
Type	BIGINT(20)	Article Type for Monitoring Server	0:Normal Article, 2:Deleted Article
Title	VARCHAR(255)	Article Headline	
URL	VARCHAR(255)	Original Article URL	text
GetTime	DATETIME	Collected Time	datetime
Body	TEXT	Article Text	text

tbmonservsitetree Site Tree Table for MonServ

Column	Type	Description	Value and Meaning
MonSiteID	BIGINT(20)	Monitoring Site ID Number	Unique ID Number
MonSitePID	BIGINT(20)	Parent MonSiteID Number	ID Number
Type	BIGINT(20)	Tree Node Type	0:Folder, 1:Site
MonSiteName	VARCHAR(255)	Monitoring site Title or Folder Name	
Language	BIGINT(20)	Providing language from this site	0:English, 1:Korean, 2:Japanese
nTime	BIGINT(20)	Monitoring Time	
Status	BIGINT(20)	Monitoring Agent Status	0:Working, 1:Collecting
Description	TEXT	deacription of Monitoring Site	

tbmonservsite Site Informain Table

Column	Type	Description	Value and Meaning
OneSiteID	BIGINT(20)	Single Site ID Number	Unique ID Number
MonSiteID	BIGINT(20)	Monitoring Site ID Number	ID Number
MonSiteURL	VARCHAR(255)	Single Site Original Site	
Method	BIGINT(20)	Sending Data Method	
Enctype	BIGINT(20)		
PurposeType	BIGINT(20)		
SetMonitoring	BIGINT(20)		
Sequent	BIGINT(20)		

tbmonservuserdetail User Information Table

Column	Type	Description	Value and Meaning
ID	BIGINT(20)		
UserID	VARCHAR(16)		
Passwd	VARCHAR(41)		Encrypted by mySQL PASSWORD() Function
FirstName	VARCHAR(255)		
SecondName	VARCHAR(255)		
Email	VARCHAR(255)		
JoinDate	DATETIME		
Availability	ENUM('Y','N')		

tbmonservuserdomain User and Domain Relation Table

Column	Type	Description	Value and Meaning
UserID	VARCHAR(16)		
DomainName	VARCHAR(255)		
Privilege	BIGINT(20)		
LoginStatus	ENUM('Y','N')		
SuperUser	ENUM('Y','N')		
RegisterDate	DATETIME		

tbmonservdomainenv Domain Information Table

Column	Type	Description	Value and Meaning
DomainID	BIGINT(20)		
DomainName	VARCHAR(255)		
RegisterDate	DATETIME		

For mySQL DB Connection

mySQL Server Address	seoul.comp.utas.edu.au
mySQL User Name	sspark
mySQL User Password	**********
mySQL Database Name	iWebKXA459

For Classification

Domain	sspark
User Name	sspark
User Password	**********

*** Please, Do not miss schema of Classifier Table (next sheet), you need it more than this sheet.**

(2) Tables used by MCRDR-Classifier

Domain is your user name
ex) my user name is sspark
so, my table is tbclassify_sspark_article
tbclassify_sspark_articlefolder
....and so on

tbclassify_DOMAIN_article Domain Article indexing Table

Column	Type	Description	Value and Meaning
ArticleID	BIGINT(20)	Article ID Number	
TypeID	BIGINT(20)	Article Attribute	0:Unclassified Article, 1:Classified Article, 2:Deleted Article (Trash bin)
bRead	BIGINT(20)	Article Read,Unread for MonClassifier	
Kwd	VARCHAR(255)	DO NOT USE (Meaning is nothing)	
ClassifiedDate	DATETIME	DO NOT USE (Meaning is nothing)	
HitCount	BIGINT(20)	From Web Hit Count	
TopNews	ENUM('Y','N')	Selected by web admin for main headline article	Y:top news article, N:just nomal article
ImgFileName	VARCHAR(255)	Added by web admin	if Article including a picture, image file name
ImgType	VARCHAR(255)	Added by web admin	if Article including a picture, image file name

sIMG	TEXT	Added by web admin	image file data
bIMG	TEXT	Added by web admin	image file data

tbclassify_DOMAIN_articlefolder Relation table between Article and Folder

Column	Type	Description	Value and Meaning
ArticleID	BIGINT(20)	Article ID Number	
FolderID	BIGINT(20)	Folder ID Number	
ManualClassified	ENUM('Y','N')		Y: Classified Article by MonClassifier manually, N:Classified Article Automatically
ClassifiedDate	DATETIME	Classified Date and Time	

tbclassify_DOMAIN_foldertree Folder Tree Table

Column	Type	Description	Value and Meaning
FolderID	BIGINT(20)	Folder ID	Folder ID Number
FolderPID	BIGINT(20)	Parent Folder ID	Folder Parent ID Number
FolderName	VARCHAR(255)	Folder Name	Folder Name
Description	VARCHAR(255)	Folder Description	
RegisterDate	DATETIME	Created datetime	
Enable	ENUM('Y','N')	Folder Enable or not	Y:Enable, N:Disable
Attribute	ENUM('N','E','X')	DO NOT USE(attribute for folder)	

tbclassify_DOMAIN_monsitekwd (DO NOT USE) Site Keyword Table

Column	Type	Description	Value and Meaning

tbclassify_DOMAIN_monsitekwdtree MonClassifier Site Tree

Column	Type	Description	Value and Meaning
ItemID	BIGINT(20)	Tree Item Node ID	Node Item ID number
ItemPID	BIGINT(20)	Tree Item Node Parent ID	
ItemName	VARCHAR(255)	Node Name	Item Name
Description	VARCHAR(255)	Node Description	
ItemType	BIGINT(20)	Node Item Type	5:Folder, 6:Site
MonSiteID	BIGINT(20)	Monitoring Site ID	Site ID in Monitoring Server
PauseCollect	ENUM('Y','N')	Stop providing article	Y: Stop providing article from this site, N: getting article from this site continually
ClassifyType	BIGINT(20)	Classification Mode	0:Don't Classification, 1:Auto Classification

tbclassify_DOMAIN_rulecondition Condition Table with Rule relation

Column	Type	Description	Value and Meaning
ConditionID	BIGINT(20)	Condition ID	Condition ID
Condition	VARCHAR(255)	Condition Name	Single condition Keyword
ConType	BIGINT(20)	Condition Type	0:Keyword in the Head, 1:Keyword in the Body, 2:Keyword in Anywhere
RuleID	BIGINT(20)	Rule ID	
RegisterDate	DATETIME	Register Date and Time	

tbclassify_DOMAIN_ruletree Rule Tree Table

Column	Type	Description	Value and Meaning
RuleID	BIGINT(20)	Rule ID Number	
RulePID	BIGINT(20)	Rule Parent ID Number	

RuleType	BIGINT(20)	Rule Type	0: Nomal rule, 1: Stop Rule(doesn't have conclusion folder ID)
ConclusionID	BIGINT(20)	Conclusion Folder ID Number	
CornerstoneCaseID	BIGINT(20)	Article ID	when you create a rule with this article
RegisterDate	DATETIME	Register Date and Time	register date and time